

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
Кафедра Комп'ютерних інтелектуальних технологій та систем

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Короткострокове прогнозування часових рядів із застосуванням штучних
нейронних мереж

Виконала:

студентка 2 курсу, групи КІТм-21-1

Пасічник К. Ю.

(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма Комп'ютерні
інтелектуальні технології

Керівник проф. Руденко О. Г.

(посада, прізвище, ініціали)

Допускається до захисту

(підпис)

Зав. кафедри КІТС

(підпис)

О. Г. Руденко

2022 р.

Харківський національний університет радіоелектроніки

Факультет	Комп'ютерної інженерії та управління
Кафедра	Комп'ютерних інтелектуальних технологій та систем
Рівень вищої освіти	другий (магістерський)
Спеціальність	123 Комп'ютерна інженерія
Тип програми	освітньо-професійна
Освітня програма	Комп'ютерні інтелектуальні технології

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Пасічник Катерини Юріївни
(прізвище, ініціали)

1. Тема роботи (проекту) _____ Короткострокове прогнозування часових рядів із
застосуванням штучних нейронних мереж

затверджена наказом по університету від « 7 » листопада 2022 р. № 1455Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 10 грудня 2022 р.

3. Вхідні дані до роботи (проекту) _____ Теоретичні відомості про існуючі методи
прогнозування та ШНМ. Складові частини часового ряду. Застосування прогнозування
часових рядів в ШНМ. Набір даних про щомісячні показники концентрації CO₂ в
атмосфері. Аналіз та прогнозування рівня вуглекислого газу в атмосфері за допомогою
ШНМ

4. Перелік питань, що потрібно опрацювати в роботі
Аналіз існуючих методів прогнозування

Огляд типів штучних нейронних мереж

Штучні нейронні мережі в задачах прогнозування часових рядів

Аналіз набору даних рівня вуглекислого газу в атмосфері за певний період часу

Програмна реалізація прогнозування

Тестування системи

Висновки по роботі

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням кафедри)

14 слайдів

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно до наказу, зазначеному у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Видача та узгодження теми проекту	7.11.2022	виконане
2	Аналіз предметної області, постановка задачі, вибір інструментальних засобів	8.11.2022 – 10.11.2022	виконане
3	Розробка алгоритмів та структури роботи	11.11.2022 – 17.11.2022	виконане
4	Збір та аналіз даних для прогнозування	18.11.2022 – 21.11.2022	виконане
5	Розробка алгоритму прогнозування	21.11.2022 – 27.11.2022	виконане
6	Проведення тестування системи	28.11.2022 – 30.11.2022	виконане
7	Оформлення пояснювальної записки	01.12.2022 – 16.12.2022	виконане
8	Перевірка виконаного проекту керівником	17.12.2022	виконане
9	Захист проекту	20.12.2022	виконане

Дата видачі завдання 7 листопада 2022 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

проф. Руденко О. Г.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної магістерської роботи містить: 95 ст., 23 рис., 1 дод., 22 джерела.

ЧАСОВІ РЯДИ, ШТУЧНІ НЕЙРОННІ МЕРЕЖІ, RNN, TCN, ПРОГНОЗУВАННЯ, АТМОСФЕРНИЙ CO₂, PYTHON, DARTS, MATPLOTLIB, PANDAS

Метою кваліфікаційної роботи є аналіз, дослідження та прогнозування часових рядів разом зі штучними нейронними мережами (ШНМ). Часові ряди представляють великий практичний інтерес для багатьох дисциплін. Їх використання дозволяє з певною похибкою виявляти майбутні значення серії подій за попередніми значеннями. Що стосується штучних нейронних мереж, то можна впевнено сказати про їх застосування у багатьох областях статистики, однією з яких і є прогнозування часових рядів.

Окрім того, що сьогодні представлено багато успішних рішень у різних галузях, таких як економіка, фінанси, бізнес, навколишнє середовище та інші, в даній роботі буде розглянуто та реалізовано короткострокове прогнозування рівня вуглекислого газу (CO₂) в атмосфері в період боротьби з глобальним потеплінням та зміною клімату, тому тематика роботи є актуальною.

У ході виконання кваліфікаційної роботи були досліджені основні методи прогнозування, моделювання структур часових рядів за допомогою ШНМ для прогнозування рівня вуглекислого газу в атмосфері.

Результатом кваліфікаційної роботи має бути аналіз методів прогнозування на прикладі набору даних про рівень вуглекислого газу в атмосфері, актуальність обраних методів та сфери подальшого розвитку. Також очікується виконати огляд типів ШНМ та можливі методи застосування їх в прогнозуванні.

ABSTRACT

Course project: 95 pages, 23 figures, 1 appendix, 22 sources.

TIME SERIES, ARTIFICIAL NEURAL NETWORKS, RNN, TCN, FORECASTING, ATMOSPHERIC CO₂, PYTHON, DARTS, MATPLOTLIB, PANDAS

The major goal of this qualification project is a detailed analysis and study of time series together with artificial neural networks (ANNs). In many disciplines, time series are used in practice. With its help, it is possible to detect future values of an events series based on previous values with a certain error. As for artificial neural networks, they are used in many areas of statistics, one of which is time series forecasting. Today many successful solutions are presented in various fields, such as economics, finance, business, environmental studies and others, in this project will be considered and implemented the forecasting of the level of carbon dioxide in the atmosphere in times of struggling against global warming and climate changes. That is why this topic is relevant.

During the qualification project the main methods of forecasting, modeling of time series structures using ANN for forecasting the level of carbon dioxide in the atmosphere were studied.

The result of the qualification project should be an analysis of forecasting methods with an example of a set of data on the level of carbon dioxide in the atmosphere, the relevance of the chosen methods and areas of further development. It is also expected to review the types of ANN and possible methods of their application in forecasting.

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет
Кафедра

Комп'ютерної інженерії та управління
Комп'ютерних інтелектуальних технологій та систем

АНОТАЦІЯ

КВАЛІФІКАЦІЙНОЇ РОБОТИ

рівень вищої освіти

другий (магістерський)

Короткострокове прогнозування часових рядів із

застосуванням штучних нейронних мереж

(тема)

Виконала:

студентка 2 курсу, групи КІТм-21-1

Пасічник К. Ю.

(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма Комп'ютерні інтелектуальні
технології

Керівник

проф. Руденко О. Г.

(посада, прізвище, ініціали)

2022 р.

Пасічник К. Ю. Короткострокове прогнозування часових рядів із застосуванням штучних нейронних мереж. – Магістерська кваліфікаційна робота.

У магістерській кваліфікаційній роботі вирішено актуальну задачу прогнозування рівня вуглекислого газу в атмосфері в частках на мільйон за обраний період часу.

Метою кваліфікаційної роботи є аналіз, дослідження та прогнозування часових рядів разом зі штучними нейронними мережами (ШНМ), а також практичне застосування отриманих знань для передбачення рівня вуглекислого газу в атмосфері у найближчому майбутньому на основі набору даних з відкритого джерела за попередні роки. Часові ряди є одним із найпопулярніших підходів до прогнозування розвитку різноманітних галузей та процесів: від метеорології, економетрики та фінансів до питань з навколишнього середовища та енергетики, саме тому цей напрямок розглянуто та описано в даній кваліфікаційній роботі. Завдяки додатковим можливостям запису даних і їх аналізу у людства є також можливість вивчати часові ряди для аналізу промислових процесів або відстеження бізнес-метрик, таких як продажі.

Об'єктом дослідження цієї роботи є вивчення області комп'ютерних інтелектуальних технологій та машинного навчання, а саме – нейронних мереж, їх класифікацій, властивостей та застосування разом з часовими рядами. Оскільки ШНМ може моделювати як нелінійні, так і лінійні структури часових рядів, їх використання при прогнозуванні може дати кращі результати, ніж інші методи. Як теоретичні, так і емпіричні висновки в досліджуваних проектах показують, що поєднання різних методів може бути ефективним способом покращення прогнозів

Предметом дослідження у даній роботі є методи моделювання структур часових рядів за допомогою ШНМ для прогнозування рівня вуглекислого газу в атмосфері. Розкриваючи саме цю тему, можна зрозуміти вплив надмірної кількості CO₂ на навколишнє середовище та зміни клімату в цілому.

Проблема глобального потепління була і лишається актуальною вже багато років, саме тому важливо розуміти аспекти, які безпосередньо впливають на проблему, таким чином знаходити рішення та здійснювати превентивні заходи. Прогнозування за даним предметом дослідження було здійснено за допомогою мови програмування Python та допоміжних бібліотек, розроблених спеціально для інтелектуальної обробки даних та машинного навчання, а також для якісного відображення результатів.

Нейронні мережі, штучний інтелект та інтелектуальний аналіз даних є чи не найбільш прогресуючою сферою науки у XXI столітті. Десятки років тому з активним розвитком інтернет-технологій виникла потреба автоматизувати та полегшити обробку величезних об'ємів даних, які стрімко зростали. Разом з цим людина знайшла спосіб як не лише вирішувати проблеми такого роду, але і забезпечити високу точність та якість аналізу, обробки й збереження даних, а саме – «навчити» електронні машини виконувати подібні задачі. Застосування штучного інтелекту було і є успішним у питаннях економіки, клімату, спорту, фінансів, електроніки та інших важливих сфер повсякденного життя.

Поняття прогнозування визначається також як спроба передбачити майбутнє, використовуючи при цьому якісні та кількісні методи. Методології прогнозування можна розділити на три групи, в залежності від конкретних даних, а саме:

- що відбулося до поточного моменту;
- які висновки експертів з приводу тих чи інших питань;
- конкретні дії людей.

У даній роботі важливим поняттям виступає часовий ряд – набір точок даних, які є упорядковані у часі. Обробляючи набір даних, можна виявити певну тенденцію або тренд, оскільки значення зростають протягом вибраного періоду часу. Моделювання та прогнозування часових рядів має практичне значення для різних факторів в екологічних, соціальних, економічних та інших системах. Застосування штучних нейронних мереж (ШНМ) для

вирішення такого класу задач допомагає впоратися зі складними обчисленнями та закономірностями, що не під силу лінійним методам обробки.

Серед існуючих методів прогнозування на основі штучних нейронних мереж було обрано три методи, а саме:

- наївна модель прогнозування;
- метод лінійної регресії;
- метод експоненціального згладжування.

Наївна модель прогнозування являється базовою серед моделей та зазвичай не може ефективно спрогнозувати якусь задачу на далеке майбутнє.

Методи з регресією краще можуть впоратися з такими задачами, проте досвід роботи з даною моделлю прогнозування показав, що вона так само не може бути використана в довгостроковій перспективі, натомість може застосовуватися в прогнозах для короткострокових задач.

Згладжуючі методи або експоненціальні методи згладжування можуть здійснювати процес підсумовування значень за попередні періоди часу, яким потім присвоюються вагові коефіцієнти. Зважування проводиться таким чином, що щойно отриманим даним призначаються великі ваги, а застарілим – менші. Метод експоненціального згладжування зазвичай застосовують у задачах з відносно стаціонарними умовами, тобто без різких змін умов зовнішнього та внутрішнього середовища.

Для порівняння, у кваліфікаційній роботі був проведений аналіз нейронних мереж, процес навчання нейронної мережі та застосування їх на практиці для вирішення задачі прогнозування, що описується в проекті. Штучна нейронна мережа (ШНМ) – це математична модель, побудована на принципах організації та функціонування біологічних нейронних мереж – нейронів. Для ефективного обробки даних, ШНМ проходить навчання за допомогою різних алгоритмів. Модель нейрона штучної мережі складається з множини нейронів, множини зв'язків та містить три шари: вхідний, прихований, яких може бути декілька, та вихідний. За способом навчання,

ШНМ поділяються на моделі навчання з учителем та без учителя. В даній роботі більш детально були розглянуті саме моделі навчання з учителем, а саме: багатошарові повнозв'язні нейронні мережі, згорткові нейронні мережі та рекурентні нейронні мережі. На практиці було проведено навчання темпоральної згорткової нейронної мережі (TCN) та рекурентної нейронної мережі (RNN).

Мережа TCN є похідною від класичної згорткової нейронної мережі, і призначена для роботи з часовими рядами. У основі TCN лежить операція причинно-наслідкової згортки, тобто вихідне значення залежить тільки від значень, які розташовані раніше у вхідній послідовності. Серед переваг часової згорткової нейронної мережі виділяють:

- низький рівень потреби в пам'яті для навчання;
- можливість змінювати розмір сприйнятливого поля;
- паралельна обробка одразу кількох наборів даних;
- підтримка вхідних даних змінної довжини.

Рекурентна нейронна мережа відрізняється від інших наявністю зворотних зв'язків, що дозволяє їм зберігати інформацію. У задачах часто застосовують різновид рекурентної нейронної мережі під назвою мережа з довгою короткостроковою пам'яттю (Long Short-Term Memory), тобто в один із шарів мережі закладено комірку довгої короткострокової пам'яті.

Мова програмування Python чудово підходить для глибокого навчання, оскільки в ній міститься стандартна бібліотека з безліччю модулів для машинного навчання, прогнозування та аналізу нейронних мереж.

Точність та ефективність прогнозування можна визначити за допомогою спеціальних критеріїв, таких, як середня квадратична помилка (MSE), корінь квадратний із середньої похибки прогнозу (RMSE), середня відносна похибка у відсотках (MAPE), симетрична середня відносна похибка у відсотках (SMAPE), а також коефіцієнт детермінації.

На даний момент з'являються нові, потужніші підходи до прогнозування часових рядів, які зможуть допомогти вирішувати як вузьконаправлені задачі,

так і глобальні проблеми людства. Ці підходи базуються на застосуванні таких розділів сучасної математики, як: еволюція, теорія стохастичного моделювання, теорія катастроф і теорія систем самоорганізації, включаючи генетичні алгоритми та нечітку логіку.

Результатом кваліфікаційної роботи має бути аналіз методів прогнозування та штучних нейронних мереж на прикладі набору даних про рівень вуглекислого газу в атмосфері, порівняльний аналіз досліджених методів, їх актуальність та сфери подальшого розвитку. Дане дослідження може бути використано для подальшого прогнозування рівня CO₂ у майбутньому.

Ключові слова: ЧАСОВІ РЯДИ, ШТУЧНІ НЕЙРОННІ МЕРЕЖІ, ПРОГНОЗУВАННЯ, АТМОСФЕРНИЙ CO₂, PYTHON, RNN, TCN

ПЕРЕЛІК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ РОБОТИ

1. Пасічник К. Ю. Прогнозування часових рядів із застосуванням штучних нейронних мереж. *Радіoeлектроніка та молодь у ХХІ столітті*: матеріали 25-го Міжнар. молодіжн. форуму, 20-22 квітня 2021 р. – Харків: ХНУРЕ, 2021. – Т. 5. – С. 189–190.

2. Пасічник К. Ю. Штучний інтелект та VR/AR технології. *Радіoeлектроніка та молодь у ХХІ столітті*: матеріали 25-го міжнар. мол. форуму. Харків, 2021. С. 53.

ПЕРЕЛІК ПУБЛІКАЦІЙ КЕРІВНИКА ТА СПІВРОБІТНИКІВ КАФЕДРИ, ВИКОРИСТАНИХ В РОБОТІ

1. O. Rudenko, O. Bezsonov, i O. Romanyk, Neural network time series prediction based on multilayer perceptron. *Development Management*. 2019. Vol. 17,

vol. 1. P. 23–34.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	14
ВСТУП.....	15
1 ІСНУЮЧІ МЕТОДИ ПРОГНОЗУВАННЯ	17
1.1 Поняття «часовий ряд»	17
1.2 Класифікація часових рядів.....	18
1.3 Методології прогнозування.....	20
1.3.1 Якісні методи прогнозування.....	22
1.3.2 Методи на основі часових рядів.....	23
1.4 Складові частини часового ряду	27
1.5 Введення до аналізу часових рядів	29
1.6 Критерії точності прогнозів.....	30
2 ОГЛЯД ТИПІВ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ	33
2.1 Поняття ШНМ	33
2.2 Топології нейронних мереж	37
2.2.1 Нейронні мережі прямого поширення.....	39
2.2.2 Нейронні мережі зворотного поширення.....	41
2.3 Згорткова нейронна мережа	45
2.3.1 Згортковий шар	48
2.3.2 Підвибірковий шар.....	48
2.3.3 Повнозв'язний шар	49
2.4 Темпоральна згорткова мережа для вирішення задачі прогнозування	49
2.4.1 Переваги та недоліки темпоральної згорткової нейронної мережі у порівнянні з RNN	52
2.3 Основи навчання нейронної мережі	53
3 АНАЛІЗ ІНСТРУМЕНТІВ РЕАЛІЗАЦІЇ ЗАДАЧІ ПРОГНОЗУВАННЯ	55
3.1 Середовище розробки	55
3.2 Мова програмування Python.....	56

3.2.1	Чому варто використовувати саме Python?.....	57
3.2.2	Бібліотека Darts	59
3.2.3	Бібліотека Pandas.....	60
3.2.4	Бібліотека Matplotlib	61
4	РОЗРОБКА СИСТЕМИ ПРОГНОЗУВАННЯ РІВНЯ ВУГЛЕКИСЛОГО ГАЗУ У АТМОСФЕРІ.....	62
4.1	Аналіз методів часових рядів для обраної задачі прогнозування	63
4.2	Побудова моделей прогнозування в середовищі Jupyter Notebook.....	64
4.3	Прогнозування часових рядів.....	69
4.3.1	Створення найвної моделі прогнозування	70
4.3.2	Створення моделі прогнозування експоненціального згладжування	72
4.3.3	Створення моделі прогнозування лінійної регресії.....	74
4.3.4	Створення моделі прогнозування часової згорткової мережі (TCN).	76
4.3.5	Створення моделі прогнозування на основі рекурентної нейронної мережі	78
4.4	Порівняльний аналіз отриманих результатів дослідження	81
4.5	Створення прогнозу рівня вуглекислого газу на 2023 рік	83
	ВИСНОВКИ.....	86
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	88
	ДОДАТОК А	Ошибка! Закладка не определена.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

МПП – нейронна мережа прямого поширення

ШНМ – штучна нейронна мережа

ARIMA – інтегрована модель авторегресії змінного середнього (англ., Autoregressive Integrated Moving Average)

ARMA – модель авторегресії змінного середнього (англ., Autoregressive Moving Average)

Deep Learning – глибоке навчання

MAE – середня абсолютна помилка (англ., Mean Absolute Error)

MAPE – середня відносна помилка (англ., Mean Absolute Percentage Error)

RMSE – квадрат середньої квадратичної помилки (англ., Root Mean Square Error)

RNN – рекурентна нейронна мережа (англ., Recurrent Neural Network)

SMAPE – симетрична середня відносна помилка (англ., Symmetric Mean Absolute Percentage Error)

TCN – темпоральна або часова згорткова нейронна мережа (англ., Temporal Convolutional Network)

TSDM – інтелектуальний аналіз часових рядів (англ., Time Series Data Mining)

ВСТУП

В сучасному світі новітніх технологій, де інформація стала основою для розвитку будь-якої сфери життя, добування даних (Data Mining) являється необхідною, сучасною методологією, яка стрімко набирає популярності. Data Mining – інтелектуальний аналіз даних, результат природної еволюції інформаційних технологій від систем зі збору даних і організації баз даних, через системи управління даними – до сучасних інтегрованих систем

Задачі класифікації, моделювання та прогнозування є основними для даного напрямку, але саме остання користується величезним попитом, коли справа стосується аналізу часових рядів (ЧР). Часові ряди – це послідовність упорядкованих у часі числових показників, що характеризують рівень стану і зміни досліджуваного явища.

Розвиток методів прогнозування безпосередньо пов'язаний з розвитком ІТ, зокрема, зі зростанням обсягів інформації, що зберігаються і ускладненням методів і алгоритмів прогнозування, реалізованих в інструментах Data Mining. В результаті цього виник термін, що носить назву інтелектуальний аналіз часових рядів або Time Series Data Mining (TSDM).

Моделювання та прогнозування часових рядів має практичне значення для різних факторів в екологічних, соціальних, економічних та інших системах. Застосування штучних нейронних мереж (ШНМ) для вирішення такого класу задач допомагає впоратися зі складними обчисленнями та закономірностями, що не під силу лінійним методам обробки. Методи ШНМ використовуються уже кілька десятиліть, проте з того часу інтерес до них лише росте. В еру індустрії 4.0 можливості нейромережевих технологій використовуються в багатьох галузях науки, починаючи від медицини і астрономії, закінчуючи інформатикою та економікою.

Існують і певні недоліки практичного використання ШНМ. Це, тривалий час, який витрачається на процес моделювання, і великий обсяг даних, що

вимагається сучасною технологією ШНМ. Прискорення обробки інформації досягається завдяки вражаючому прогресу у збільшенні тактової частоти сучасних процесорів. Більшість сучасних методологій використовують мережі з великою кількістю параметрів ("ваги"). Тут мається на увазі, що тривалих обчислень для встановлення їх значень та вимог до багатьох спостережень не уникнути. Саме тому на практиці доступною є лише невелика кількість оброблених даних.

Результатом кваліфікаційної роботи буде вирішення прикладної задачі прогнозування рівня вуглекислого газу в атмосфері за допомогою прогнозування часових рядів та ШНМ, що дозволить передбачити поведінку різних факторів в обраній сфері дослідження. В ході реалізації буде здійснюватися побудова, ідентифікація, налаштування та перевірка готової моделі часових рядів, а також застосування моделі RNN та TCN, що дозволить оцінювати ефективність прогнозування через лінійний часовий ряд, який слідує певному відомому статистичному розподілу, такому як нормальний розподіл.

Для досягнення поставленої мети кваліфікаційної роботи необхідно виконання наступних завдань:

- дослідити існуючі методи і моделі, які лежать в їх основі;
- дослідити існуючі перетворення вхідних даних, що подаються на вхід обраним моделям, вивчити доцільність їх застосування;
- дослідити існуючі критерії порівняння точності вибраних моделей і отриманих ними оцінок прогнозу;
- описати алгоритм застосування моделей RNN та TCN для прогнозування часових рядів;
- дослідження різних способів і методів вибору конкретних параметрів моделі, як графічних, так і візуальних;
- реалізувати алгоритм прогнозування обраної задачі на основі найкращої з обраних моделей за результатами дослідження.

1 ІСНУЮЧІ МЕТОДИ ПРОГНОЗУВАННЯ

1.1 Поняття «часовий ряд»

Часовим рядом називається впорядкована за часом як динамічна, так і статична послідовність дискретних показників або спостережень $y(t_1), y(t_2), \dots, y(t_N)$, яка характеризує рівень стану та зміни певного досліджуваного явища, яке протікає в умовах невизначеності протягом N періодів. У підручниках з загальної теорії статистики переважають терміни ряди динаміки, динамічні ряди, статистичне вивчення динаміки. У зарубіжній англомовній літературі прийнято термін «time series». Усі ці терміни найближче передаються українською як часові ряди або аналіз часових рядів [1].

На рисунку 1.1 зображений приклад стаціонарного часового ряду.

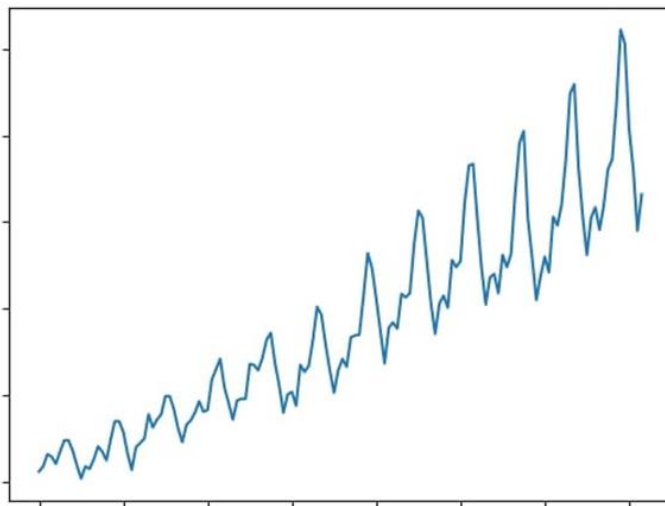


Рисунок 1.1 – Приклад часового ряду

Детерміновані процеси характеризуються достатньою кількістю інформації для визначення функціональної залежності $y = f(t)$. У випадку,

якщо процес протікає в умовах невизначеності, традиційно використовують стохастичні моделі часових рядів і вивчаються числові ряди, що містять як систематичну, так і випадкову компоненту. Для моделювання процесів у реальних складно-організованих системах, можуть бути використані гібридні моделі часових рядів, що описують стохастичні та нечіткі (лінгвістичні) типи невизначеності.

В загальному вигляді задача моделювання часових рядів формулюється наступним чином.

Нехай задані значення часового ряду $Y = \{y(1), y(2), \dots, y(N)\}$, де $y(t)$ – значення показника процесу, який досліджується, який зареєстрований на t -му такті часу ($t = 1, 2, \dots, N$). В даному випадку потрібно побудувати оцінки майбутніх значень ряду за формулою (1.1):

$$\hat{Y} = \{\hat{y}(N + 1), \hat{y}(N + 2), \dots, \hat{y}(N + \tau)\}, 1 \leq \tau \leq N, \quad (1.1)$$

де τ – горизонт прогнозування.

Основна ідея, що поєднує підходи до моделювання часових рядів, базується на виділенні систематичних (регулярних) залежностей, а також на аналізі за фіксованими критеріями отриманих залишків. Незалежно від застосовуваного методу передбачається, що закономірність змін, виявлена для певного періоду часового ряду у минулому, зберігається на обмеженому відрізку часу у майбутньому [2].

1.2 Класифікація часових рядів

Розглянемо класифікацію часових рядів. Виділяють одновимірні часові ряди, які можна отримати при фіксованій кількісній характеристиці, і багатовимірні часові ряди, що формуються при спостереженні декількох характеристик виділеного об'єкта. Часові ряди можуть бути дискретними і безперервними. У складі дискретних виділяють ряди для рівновіддалених

моментів спостереження і для довільних моментів спостереження. Також часові ряди бувають детермінованими і випадковими: перші отримують на основі значень деякої не випадкової функції, а другі є результатом реалізації деякої випадкової величини.

Проте серед усіх цих видів, слід приділити особливу увагу стаціонарним і нестаціонарним часовим рядам. Ряд деякої залежної функції $f(x)$ називають стаціонарним якщо закон розподілу ймовірностей випадкової величини $f(x)$ не залежить від параметра часу t . Якщо ж математичне сподівання функції залежить від часу, тоді ряд називають нестаціонарним [3].

Наявність і відсутність залежності ймовірнісних характеристик від часу визначає такі фундаментальні властивості процесу, як стаціонарність і ергодичність. Умова ергодичності поєднує в собі і умову стаціонарності випадкового процесу. Ергодичність проявляється в тому, що з часом процес стає однорідним.

Більшість стаціонарних випадкових процесів задовольняють властивості ергодичності. Воно проявляється в тому, що кожна реалізація випадкового процесу достатньої тривалості несе практично повну інформацію про властивості всього ансамблю реалізацій, що дозволяє істотно спростити процедуру визначення статистичних характеристик, замінюючи усереднення значень по ансамблю реалізацій усередненням значень однієї реалізації за тривалий інтервал часу. Така властивість може застосовуватися при прогнозуванні біржових курсів на різні періоди часу (короткострокові, середньострокові, довгострокові), в медицині та споживанні енергоресурсів.

Якщо розглянути динаміку перебігу реального процесу в умовах середовища, яке постійно змінюється, то можна зробити висновок, що він має властивість пристосовуватися. А модель, як основний засіб аналізу та прогнозу, в свою чергу, адаптується до ряду, який представляє цей процес. Оскільки розглядаються варіативні, нестаціонарні ряди, тобто ряди, у яких швидкість лінійного росту та інші характеристики не залишаються постійними в часі, модель буде завжди динамічною, знаходитися в русі [4].

1.3 Методології прогнозування

Поняття прогнозування може бути визначено як спроба передбачити майбутнє, використовуючи при цьому якісні та кількісні методи. Це процес дослідження, який має за мету виявити тенденції розвитку об'єкту та зовнішніх чинників. Прогнозування майбутніх значень часового ряду називається екстраполяцією, а відновлення пропущених значень називають терміном інтерполяція. Методології прогнозування умовно поділяють на три напрямки, в залежності від конкретних даних, а саме:

- що відбулося до поточного моменту (аналіз часових рядів, регресійний аналіз);
- які висновки експертів з приводу тих чи інших питань (опитування, анкетування);
- конкретні дії людей (реакційні тести) [1, 5].

Дані, отримані на підставі попередніх подій, відносно легко зібрати і проаналізувати, але вони можуть бути нерелевантні або ж майбутні значення даного процесу можуть ніяк не корелювати з попередніми на момент їх обробки. Дані, зібрані з опитувань і анкет, дорожче коштують та потребують критичної оцінки, так як наміри, демонстровані в них, не завжди приводять до якихось конкретних рішень. І, нарешті, дані, отримані і записані виходячи з конкретних дій людини, є найбільш надійними і точними, але разом з цим дуже затратними і, найчастіше, просто недоступними.

Зазвичай процес прогнозування розбивають на окремі кроки для того щоб відстежувати ефективність на кожному з них. Спочатку відбувається ідентифікація загальної потреби та мети прогнозування. Після цього відбувається етап вивчення об'єкту прогнозування та вибір методу. Далі проходить етап збору даних та передача цих даних в тому вигляді, який задовольняє умови обраної моделі прогнозування. Два останніх кроки – безпосередньо прогнозування та оцінка прогнозу [3].

Існує багато видів прогнозів, які відносяться до тієї чи іншої ознаки класифікації: період часу (короткострокові, середньострокові, довгострокові), тип прогнозування (нормативні прогнози, експертні оцінки, пошукові прогнози), за ступенем ймовірності майбутніх подій (варіантні, інваріантні), за способом подачі результатів (точкові, інтервальні). Як було зазначено, прогнози часто класифікують по тимчасовому періоду їх дії на:

- короткострокові (від кварталу до 1го року);
- середньострокові (1 – 3 роки);
- довгострокові (більше 3х років).

Під короткостроковим прогнозом, як правило, мається на увазі прогноз на один інтервал часу (в крайньому випадку на кілька). Сам же інтервал може бути будь-яким. Для прогнозу важлива не величина тимчасового інтервалу, а обсяг вибірки, її однорідність та інші параметри.

Розглянемо досліджені на даний момент якісні та кількісні методи прогнозування, які представлені на рисунку 1.2 [3, 5].

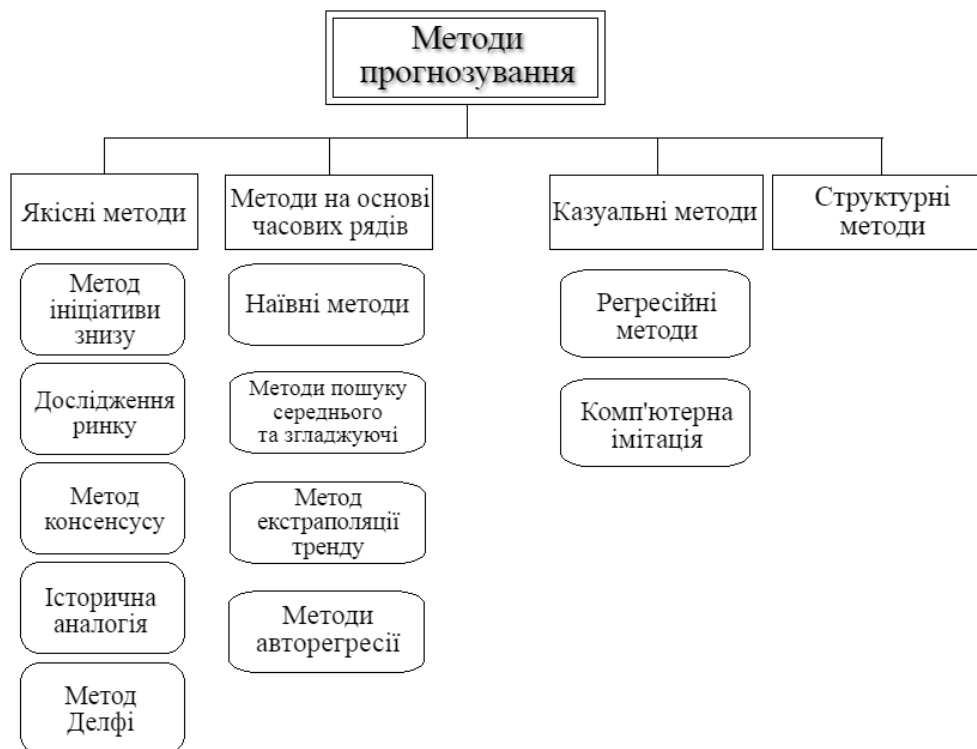


Рисунок 1.2 – Схема методів прогнозування

1.3.1 Якісні методи прогнозування

Метод ініціативи знизу будує прогноз послідовним шляхом, починаючи з самого низу. Спочатку робиться припущення, що людина, яка знаходиться ближче всього до споживача або кінцевого використання продукту більш компетентна в майбутніх потребах. Це не завжди так, але в багатьох ситуаціях дане припущення є дійсним і використовується в якості основи даного методу. Таким чином, прогнози з нижчого рівня підсумовуються і передаються на рівень вище. Процедура повторюється до тих пір, поки не досягне вищого рівня [6].

Наступний метод – дослідження ринку. Підприємства та різні фірми часто наймають сторонні компанії, які спеціалізуються на дослідженні ринку і займаються даним типом прогнозування. Вивчення ринку найчастіше застосовується для дослідження продукту з точки зору нових ідей, плюсів і мінусів існуючих рішень, аналогічних конкурентних продуктів в даному сегменті і т. д. Найчастіше, методами збору даних в цьому випадку служать опитування та інтерв'ю [7].

Ключовою ідеєю методу консенсусу є припущення, що група фахівців, котрі займають різні позиції, можуть дати більш точний прогноз, ніж маленька група осіб. В даному випадку застосовуються збори, на яких відбувається вільний обмін ідей і думок осіб усіх рівнів керівництва. Такий підхід ще носить назву «мозковий штурм». Проблемою даного методу є той факт, що люди, які займають нижчі посади, зазвичай бояться висловлювати думку на відмінну від людей з більш високими посадами [7].

Метод історичної аналогії зазвичай застосовується з урахуванням відсутності минулих даних. Наприклад, таке може статися в разі, коли організація готується до випуску нового продукту але у неї можуть вже випускатися продукти зі схожими характеристиками. В такому випадку, можна використовувати накопичені історичні дані про ці продукти. Обмеження даного підходу полягають у тому, наскільки правдиво

припущення про схожість продукту/події, а також змінюються зовнішніх факторах [7].

Останній з якісних методів – метод експертних оцінок або метод Делфі. Даний метод має схожі риси з методом консенсусу, але в даному випадку тут не має проблеми упередженості думки через дотримання анонімності обличь, які беруть участь [8]. В такому випадку кожен голос має однакову вагу. Кожному учаснику надсилається анкета з запитаннями, відповіді на яку потім додаються. Учасникам надають результати цього етапу і висилають новий опитувальник. Зазвичай, на реалізацію даного методу потрібно три ітерації.

1.3.2 Методи на основі часових рядів

У вивченні часових рядів велике місце займає питання про закономірності їх руху протягом тривалого періоду. Статистика повинна дати характеристику змін статистичних показників у часі. Найпростішим стаціонарним часовим рядом являється послідовність незалежних однаково розподілених випадкових величин. Часто показники, представлені у вигляді часового ряду, мають складну структуру, тому моделювання таких рядів шляхом побудови моделі тренда, сезонності і періодичної складової, на жаль, не дає задовільних результатів.

Найбільш поширеними моделями стаціонарних рядів є моделі авторегресії (AR) та моделі змінного рухомого середнього (IMA) з кінцевим числом параметрів. Найкращий прогноз ряду, отриманий за допомогою таких моделей, використовує кінцеве число минулих значень. Також хороший результат дає модель лінійної регресії та експоненціального згладжування.

Нарешті, варто перейти до опису методів на основі ЧР [1, 3].

1.3.2.1 Наївні методи прогнозування

Почнемо з наївних методів прогнозування. Вони вважаються найпростішими або базовими з методів прогнозування. Принцип роботи моделі полягає у тому, що для оцінки якості прогнозу використовується результуючий параметр з останнього обчислення, а всі решта попередніх значень не враховуються. Наївна модель — це тривіальне рішення, яке використовує евристику або просту статистику для обчислення прогнозу. Розробка базової моделі не завжди є точною наукою. Часто потрібна деяка інтуїція, яку ми отримуємо, візуалізуючи дані та виявляючи закономірності, які можна використовувати для прогнозування. У будь-якому проекті моделювання важливо мати базову лінію, оскільки ми використовуватимемо її для порівняння продуктивності більш складних моделей, які будемо створювати надалі. Єдиний спосіб дізнатися, чи модель виявилась ефективною, це порівняти її з базовою лінією [1, 3].

1.3.2.2 Методи змінного середнього та експоненціального згладжування

Наступними на черзі є методи змінного середнього та згладжуючі (пом'якшуючі) методи. Коли часовий ряд не змінюється занадто швидко і не має сезонних характеристик, то метод змінного середнього може бути використаний з метою видалення випадкових варіацій і здійснення оцінки прогнозу. У цьому випадку, використовується величина так званого змінного (локального) середнього. Для такої моделі дуже важливим є правильний вибір періоду для оцінки змінного середнього, так як, наприклад, при виборі великого часового періоду і наявності невеликого тренду дані будуть від нього «відставати», в той час як занадто короткий період буде надавати більшого значення різним варіаціям [10]. Отже, суть методу змінного середнього полягає в тому, що значення прогнозованого показника визначається шляхом усереднення значень за кілька попередніх моментів часу.

Ідея згладжуючих методів або методів експоненціального згладжування полягає у тому, що здійснюється процес підсумовування значень за попередні періоди часу, яким потім присвоюються вагові коефіцієнти. Зважування проводиться таким чином, що щойно отриманим даним призначаються великі ваги, а застарілим – менші. Метод експоненціального згладжування може бути використаний тільки у відносно стаціонарних умовах при стабільному розвитку системи та без різких змін умов зовнішнього та внутрішнього середовища. Наприклад, якщо компанія стала зазнавати збитків, ситуація на ринку непередбачувано змінилася, то в такому разі даний метод не рекомендується використовувати, тому що він не дозволяє врахувати вплив таких факторів [4].

1.3.2.3 Модель лінійної регресії

Щоб продемонструвати кількість підтверджених випадків через деякий час і передбачити майбутній розвиток, розглядають регресію. Щоб змодельовати зв'язок між відповіддю та пояснювальною змінною, використовують модель лінійної регресії. Проста лінійна регресія – це модель з одним регресором x , який відповідає змінній відповіді y . Просту лінійну регресію можна сформулювати наступним чином:

$$y = \beta_0 + \beta_1 x + \varepsilon, \quad (1.2)$$

де β_0 – це вільний член (перетин) лінії оцінки;

β_1 – коефіцієнт нахилу;

ε – компонент випадкової похибки.

β_0 та β_1 є постійними.

Лінійна регресія зазвичай застосовується для того, щоб мінімізувати залишкову суму квадратів між спостережуваними цілями в наборі даних і цілями, передбаченими лінійною апроксимацією. Досвід роботи з даною моделлю прогнозування показує, що вона не може бути використана в

довгостроковій перспективі, але може забезпечувати кращий прогноз для короткострокових задач [3].

1.3.2.4 Авторегресійна модель та модель ARIMA

Модель авторегресії (Autoregressive model, AR) – модель часових рядів, в якій значення часового ряду в даний момент лінійно залежать від попередніх значень цього ж ряду. Авторегресійна модель призначена для опису стаціонарних часових рядів і випадкових систем, що володіють інерцією. Стаціонарний процес задовольняє рівняння авторегресії нескінченного порядку з коефіцієнтами, котрі швидко зменшуються. Тому авторегресійна модель досить високого порядку може добре апроксимувати майже будь-який стаціонарний процес [1].

Розширенням для моделі прогнозування змінного середнього (MA) та авторегресії (AR) є модель авторегресії змінного середнього, (Autoregressive moving average, ARMA) – нестаціонарна лінійна математична модель аналізу та прогнозування стаціонарних часових рядів, що є узагальненням моделі авторегресії та моделі змінного середнього.

Існує також модель авторегресії інтегрованого змінного середнього (Auto Regressive Integrated Moving Average, ARIMA), яка є узагальненням моделі ARMA і містить три параметри (p, d, q) , де параметр p – порядок авторегресійної частини моделі, параметр d або ступінь диференціювання позначає кількість відмінностей у вхідних спостереженнях, параметр q є оператором сезонної авторегресії та ковзного середнього. Цю модель ще називають моделлю Бокса і Дженкінса, оскільки саме вони запропонували та розвивали дану модель.

Така модель може інтерпретуватися як лінійна модель множинної регресії, в якій в якості основних змінних виступають минулі значення самої залежної змінної, а в якості регресійного залишку – середні змінні з елементами білого шуму. ARMA-процеси мають більш складну структуру в порівнянні зі

схожими за поведінкою AR- або MA-процесами в чистому вигляді, але при цьому ARMA-процеси характеризуються меншою кількістю параметрів, що є однією з їхніх переваг [1].

Для побудови моделі ARMA за серією спостережень необхідно визначити порядок моделі, а потім і самі коефіцієнти. Для визначення порядку моделі може застосовуватися дослідження таких характеристик часового ряду, як його автокореляційна функція і приватна автокореляційна функція. Для визначення коефіцієнтів застосовуються такі методи, як метод найменших квадратів і метод максимальної правдоподібності, але в даному випадку розглядати математичну складову такої моделі недоцільно. Як показує практика, модель ARIMA перевершує інші моделі прогнозування часових рядів, тому що вона поєднала у собі авторегресію ковзного середнього, модель аналізу часових рядів та методологію.

1.4 Складові частини часового ряду

У вивченні часових рядів велике місце посідає питання про закономірності їх руху протягом довгого періоду часу. Типовий часовий ряд є сумою детермінованого та випадкового параметрів (1.3):

$$y_t = d_t + r_t, t = \overline{1, T}, \quad (1.3)$$

де d_t – детермінований параметр;

r_t – випадковий або нерегулярний параметр.

Детермінований параметр складається з трьох основних частин (1.4):

$$d_t = t_t + s_t + c_t, t = \overline{1, T}, \quad (1.4)$$

де t_t – тренд (стійка систематично змінна довготривала складова);

s_t – сезонна складова;

c_t – циклічні коливання [3].

Трендом, або основною тенденцією, називається характеристика процесу зміни якогось явища за тривалий час без випадкових коливань, що створюються другою групою факторів. Трендом ще називають детерміновану основу моделі, яка є змінною в залежності від часу. Тенденція існує, коли спостерігається тривале збільшення або зменшення даних, тобто вона не має бути лінійною. Наприклад, серії, що стосуються приросту населення, кількості будинків у місті тощо, демонструють тенденцію зростання, тоді як тенденцію зниження можна спостерігати у серіях, що стосуються рівня смертності, епідемій тощо.

Циклічна складова проявляється протягом тривалого часу і є результатом дії факторів, що володіють великою післядією або циклічно змінюються з часом. Цикл виникає, коли множина даних демонструють зростання та спад, що не мають фіксованої частоти. Характерним прикладом служать цикли ділової активності, демографічні та астрофізичні цикли. Тривалість циклу триває довший проміжок часу, як правило, два або більше років. Більшість економічних та фінансових часових рядів демонструють певні циклічні коливання. Наприклад, діловий цикл складається з чотирьох фаз, а саме: процвітання, занепад, депресія та відновлення [1].

Сезонна складова обумовлена дією деякого механізму, що періодично повторюється в певний час сезону або пори року. Важливими факторами, що спричиняють сезонні коливання, є: клімат та погодні умови, традиційні звичаї тощо. Наприклад, продажі морозива зростають влітку, продажі вовняних тканин зростають взимку [1, 4].

Випадкова або нерегулярна складова не піддається обліку і реєстрації, утворена в результаті суперпозиції великого числа зовнішніх факторів, що не беруть участь у формуванні детермінованої складової. Ці зміни спричинені такими випадками, як війна, страйк, землетрус, повінь, революція тощо. Не існує визначеної статистичної методики вимірювання випадкових коливань у часових рядах.

Середніми показниками часових рядів являються середній рівень ряду, середнє абсолютне прискорення та зміна, середні темпи росту. Всі ці показники характеризують тенденції. Вони необхідні при узагальненні показників тенденції за тривалий період, за різними періодами; вони незамінні у порівнянні розвитку за неоднакові за тривалістю відрізки часу і при виборі аналітичного висловлювання тренду. За наявності динамічному ряду істотних коливань рівнів визначення середніх показників тенденції вимагає застосування спеціальних методів статистики. У цьому розділі розглядаються лише форма, математичні характеристики середніх показників динаміки і найпростіші прийоми їх обчислення, застосовні практично до рядів зі слабкою коливанням.

Таким чином, відповідно до формули (1.5), часовий ряд розглядають як суму чотирьох основних компонентів:

$$y_t = t_t + s_t + c_t + r_t, t = \overline{1, T}. \quad (1.5)$$

1.5 Введення до аналізу часових рядів

На практиці відповідна модель пристосовується до конкретного часового ряду, а відповідні параметри оцінюються з використанням відомих значень даних. Процедура налаштування часових рядів під відповідну модель називається аналізом часових рядів [8]. Він включає методи, які намагаються зрозуміти природу серії та часто корисні для майбутнього прогнозування та моделювання.

Під час прогнозування часових рядів збираються та аналізуються попередні спостереження для розробки відповідної математичної моделі, яка фіксує основний процес формування даних для ряду [8]. Потім за допомогою моделі прогнозуються майбутні події. За графіком часового ряду можна визначити наявність тренду та його характер, сезонні та циклічні компоненти, а також ступінь коливання змін послідовних значень часового ряду без

застосування тренду. В подальшому в цій роботі необхідною умовою буде визначення основних компонентів часового ряду, а саме: тренду, сезонності та залишкової компоненти. Залишки аналізують для перевірки адекватності моделі та побудови надійних інтервалів.

Зазвичай на останньому етапі аналізу часового ряду проводиться екстраполяція або інтерполяція даних і визначення рівня точності прогнозу на основі обраної моделі. Щоб уникнути неоднозначності при виборі моделі, потрібно здійснити декілька прогнозів за допомогою різних методів прогнозування.

Такий підхід особливо корисний, коли не існує великих знань про статистичну модель, за якою слідує послідовні спостереження, або коли бракує задовільної пояснювальної моделі. Прогнозування часових рядів має важливе застосування в різних сферах. Часто цінні стратегічні рішення та запобіжні заходи приймаються на основі результатів прогнозу. Таким чином, створення хорошого прогнозу, тобто підбір адекватної моделі до часового ряду, є дуже важливим процесом.

1.6 Критерії точності прогнозів

Верифікація та аналіз якості побудованої моделі часових рядів відповідно до системної методології у статистичному підході засновані на проведенні імітаційних експериментів, у процесі яких реалізується багатокритеріальне оцінювання якості моделі. При цьому визначаються внутрішні та зовнішні параметри якості моделі на основі стандартизованих критеріїв. До таких критеріїв відносять насамперед критерії точності моделювання, що дозволяють порівнювати конкуруючі моделі часових рядів [2].

Мірою точності або адекватності прогнозу є розмір різниці між фактичним і прогнозованим значеннями досліджуваних даних. Щоб визначити

таку різницю існує багато критеріїв, які допоможуть визначити точність прогнозу тієї чи іншої моделі. Розглянемо деякі з них.

Одним із важливих критеріїв точності моделі та її сумісності є коефіцієнт Тейла, який рахується через середньоквадратичне значення похибки прогнозу та описується формулою (1.6):

$$U = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^N (y_i)^2 + \frac{1}{N} \sum_{i=1}^N (\hat{y}_i)^2}}. \quad (1.6)$$

Значення U може дорівнювати числу в діапазоні від 0 до 1: якщо U дорівнює 1, то це свідчить, що модель не може бути використана для прогнозу, а якщо коефіцієнт U рівний 0, у такому випадку модель чудово підібрана для даної задачі прогнозування.

Коефіцієнт детермінації R^2 – це квадрат множинного коефіцієнта кореляції, який чисельно показує частку варіації залежної змінної, яку можна передбачити на основі незалежної змінної. Коефіцієнт детермінації обчислюється за формулою (1.7):

$$R^2 = 1 - \frac{RSS}{TSS}, \quad (1.7)$$

де RSS – це сума квадратів залишків регресії;

TSS – загальна сума квадратів [9].

Він використовується в контексті статистичних моделей або, в нашому випадку, для прогнозування за допомогою часових рядів, основною метою яких є передбачення майбутніх результатів або перевірка гіпотез на основі іншої пов'язаної інформації. Він визначає, наскільки добре спостережувані результати відтворюються обраною моделлю на основі частки загальної варіації результатів, що пояснюються цією моделлю. З формули (1.7) можна

зробити висновок, що модель прогнозування є адекватною якщо коефіцієнт детермінації буде близьким до одиниці: $R^2 \rightarrow 1$. Цей показник також особливо часто використовують для порівняння кількох моделей та вибору найкращої.

У даній роботі були використані найбільш поширені абсолютні критерії адекватності моделей прогнозів на основі часових рядів, які представлені у таблиці 1.1.

Таблиця 1.1 – Абсолютні критерії точності моделей часових рядів

Критерій	Формула
Середня квадратична похибка	$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$
Корінь квадратний із середньої квадратичної похибки прогнозу	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$
Середня відносна похибка у відсотках	$MAPE = \frac{1}{n} \sum_{i=1}^n \left \frac{y_i - \hat{y}_i}{y_i} \right \times 100\%$
Симетрична середня відносна похибка у відсотках	$SMAPE = \frac{1}{n} \sum_{i=1}^n \left \frac{y_i - \hat{y}_i}{(y_i + \hat{y}_i)/2} \right \times 100\%$

Показники точності моделі прогнозування засновані на даних часових рядів, що використовуються для генерування моделі. Усі показники точності ґрунтуються на хронологічних даних. Показники точності можуть також використовуватися як індикатор точності прогнозу, але на майбутні значення вони не переносяться.

2 ОГЛЯД ТИПІВ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

2.1 Поняття ШНМ

Штучна нейронна мережа – це математична модель, побудована на принципі організації та функціонування біологічних нейронних мереж – нервових клітинних мереж живого організму. З погляду математики це система пов'язаних і взаємодіючих обчислювальних одиниць, які називаються нейронами (рисунок 2.1). З'єднані між собою нейрони і утворюють ШНМ. З того моменту, як у 1943 році світ побачив статтю Маккалоха та Пітса про структуру штучної нейронної мережі, багато чого змінилося в розвитку нейронних мереж. На теперішній момент мало хто не знає про таку методологію, а все більше компаній, підприємств зацікавлені в залученні новітньої технології до вирішення своїх задач та проблем. Звичайно, ШНМ не можуть не зацікавити й науково-практичну сферу, де попит на розвиток засобів для полегшення праці людини тільки зростає [1, 10].

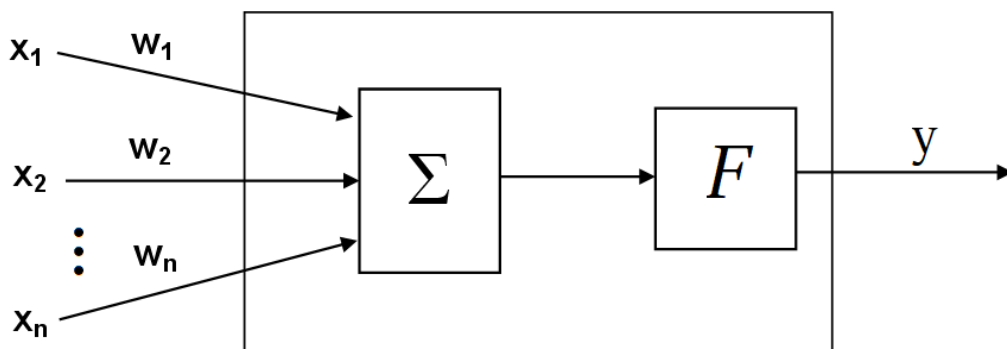


Рисунок 2.1 – Штучний нейрон

На рисунку 2.1 схематично зображено елементи штучного нейрону, де x_1, x_2, \dots, x_n – це множина входних сигналів. Кожен входний сигнал в свою чергу множиться на відповідний модифікатор зсуву або вагу w_1, w_2, \dots, w_n ,

після чого отримані значення подаються на суматор Σ , проходять через функцію активації F і перетворюються в результуючий сигнал y . Математично даний малюнок виглядає так (2.1):

$$y = F(\sum_{i=1}^n w_i x_i) = F(WX), \quad (2.1)$$

де $X = (x_1, x_2, \dots, x_n)^T$ – вектор вхідних сигналів;

$W = (w_1, w_2, \dots, w_n)$ – вектор вагових коефіцієнтів;

F – функція нелінійного перетворення.

Одним із найважливіших завдань при застосуванні нейромережевих методів є побудова мережі. Для цього потрібно виконати два етапи:

- обрати конфігурацію мережі;
- підібрати ваги для навчання мережі.

Для першого етапу важливо визначитися з типом нейронів, як між собою їх з'єднати та які входи та виходи передбачаються для майбутньої мережі. Можна сказати, що завдяки наявності безлічі різних нейромережевих архітектур, користувачам немає потреби створювати нову і дана задача зводиться до вибору найоптимальнішої серед вже існуючих. Часто вибір падає на багат шаровий персептрон, нейромережу із лінійною регресією, мережі Кохонена, мережі Хопфілда або мережі Хеммінга.

Що стосується другого етапу, то важливо підібрати такі значення ваг, щоб мережа запрацювала відповідно до ваших потреб. Кількість ваг може коливатися від десятків до тисяч, тому навчання мережі достатньо складний та трудомісткий процес. Для того щоб досягнути бажаних результатів, навчання мережі може відбуватися під контролем на кожному з етапів, тобто порівнянні очікуваних та фактичних даних, або ж бути неконтрольованим, коли мережа адаптується до певних умов та не застосовує зовнішні впливи для корекції своєї роботи.

Для того щоб обробка виконувалася ефективно, мережа проходить навчання за допомогою різних алгоритмів. За аналогією з біологічним

нейроном будується модель нейрона штучної мережі, яка складається з множини нейронів, множини зв'язків та містить три шари: вхідний, прихований, яких може бути декілька, та вихідний (рисунок 2.2).

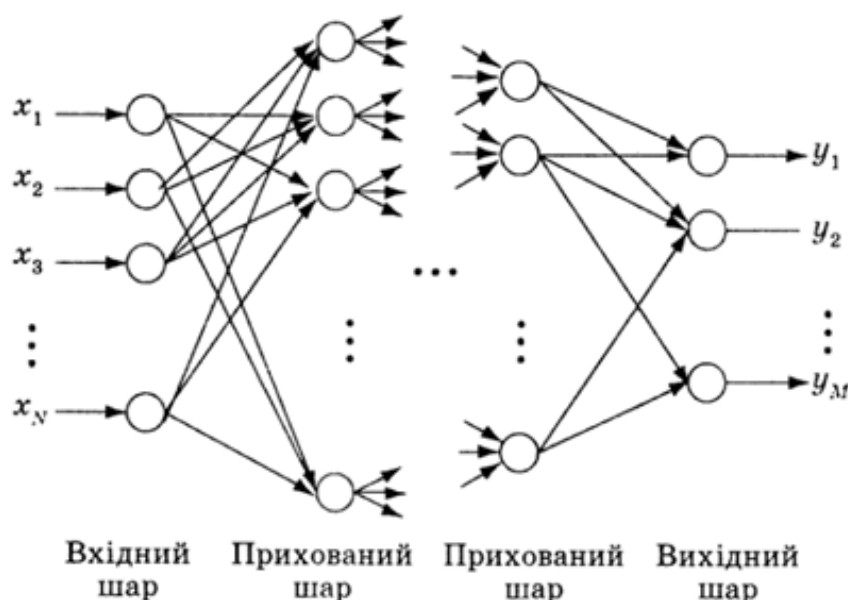


Рисунок 2.2 – Спрощена структура ШНМ

Вхідний шар структури отримує зовнішні сигнали, вихідний – відображає реакцію нейронів на комбінації вхідних сигналів, а прихований призначений для навчання поданих вхідних даних. ШНМ з прихованим шаром ще називають багатошаровими й вважаються справді нейронними, оскільки саме за рахунок наявності цього шару відбувається процес навчання. Навчання нейронної мережі зводиться до налаштування ваг синаптичних каналів. Ваги зручно подавати у вигляді елементів матриці, тому математично зв'язки між нейронами задаються у вигляді векторів та матриць [4].

В цілому, виділяють три галузі досліджень, які стосуються нейронних мереж. Перша галузь – експериментальна, заснована на фізіології та молекулярній біології. Це і не дивно, бо нейронна мережа моделює спосіб обробки інформації мозком. Наступна область – це інженерні програми нейронних мереж, розробники яких надихнулися обчисленням у стилі роботи

мозку, де інформація розподіляється як сигнал аналогового шаблону, паралельні обчислення є домінуючими, а відповідне навчання гарантує гнучкість та надійне обчислення. Один із головних напрямків в цій галузі є статистика, що відкрила нові практичні методи розпізнавання зразків, аналізу часових рядів, обробки зображень тощо. Третя область стосується математичних основ нейронних обчислень, які займаються пошуком фундаментальних принципів паралельно розподілених інформаційних систем з можливостями навчання [1, 4, 10].

Часто, коли справа стосується ШНМ, згадують про поняття глибокого навчання (Deep Learning) та машинного навчання (Machine Learning). Глибоке навчання це область машинного навчання, в якій розглядаються методи вирішення задач штучного інтелекту з використанням нейронних мереж. Ці галузі мають певний зв'язок між собою та поступово включаються одна в іншу.

Deep Learning охоплює, мабуть, найбільше коло завдань, оскільки дана технологія активно застосовується в різноманітних областях науки та техніки, має багато перспективних проектів та інновацій. Розпізнавання людської мови, автоматичне керування автомобілем, комп'ютерний зір, детектування та сегментація об'єктів – все це результати впровадження глибокого навчання в діяльність людини. Deep Learning класифікують за способами навчання: з учителем та без учителя.

Моделі навчання з учителем поділяються на багат шарові повнозв'язні нейронні мережі, згорткові нейронні мережі та рекурентні нейронні мережі. Дана група методів вимагає для визначення параметрів мережі наявність розмічених тренувальних наборів даних великих обсягів. Як правило, розмір наборів даних на порядки перевищує розміри тренувальних баз, які використовуються для навчання традиційних моделей машинного навчання. Ці моделі розрізняються способами зв'язку нейронів, що знаходяться на сусідніх шарах мережі, і кількістю нейронів попереднього шару, що впливають на стан окремих нейронів поточного шару. При цьому для

початкової ініціалізації ваг цих моделей (попереднього навчання) можуть використовуватися моделі другої групи [10].

До способів навчання без учителя відносять такі методи, як: автокодувальники, обмежені машини Больцмана, глибокі машини Больцмана, генеративно-змагальні мережі. На відміну від попередньої групи моделей, тут не вимагається наявність розмічених тренувальних наборів даних для налаштування параметрів мережі. Їхня задача – в процесі навчання тим чи іншим чином відновити вихідний сигнал, що прийшов на вхід мережі щоб мати змогу навчатися.

Всесвітньо відомі компанії-гіганти Google, Tesla, Uber та інші уже показали світу можливості інтелектуальних технологій на прикладі онлайн-перекладача Google Translate, голосових запитів на пошук будь-якої інформації в мережі Інтернет, автопілотування автомобіля, розумний розподіл замовлень таксі на базі мультиагентних систем та багато іншого.

2.2 Топології нейронних мереж

З точки зору біології, основний елемент нервової системи людини – це нервова клітина або нейрон. Для спеціаліста в області інтелектуального аналізу даних нейрон – це одиниця інформації для нейронної мережі, що отримує вхідні сигнали через поєднання з іншими нейронами, майже як і наша нервова система. Проте на відміну від системи нейронів людини, штучна нейронна мережа може мати різну структуру, особливості нейронів та має здатність навчатися.

В залежності від наявності або відсутності зворотних зв'язків в ШНМ, розрізняють наступні топології:

- ШНМ без зворотних зв'язків (прямого поширення);
- ШНМ зі зворотними зв'язками (зворотного поширення, рекурентні).

На рисунку 2.3 можна розглянути класифікацію існуючих архітектур штучних нейронних мереж.



Рисунок 2.3 – Архітектура ШНМ

Нейронна мережа прямого поширення (МПП, Feedforward neural network) – це тип нейронної мережі, в якій сигнали поширюються лише в одному напрямку, починаючи від вхідного шару нейронів, через приховані шари до вихідного шару. Результат опрацювання сигналу подається також на вихід, а зворотні зв'язки відсутні [12].

Мережами зворотного поширення або рекурентними (МЗП, РНМ, Backpropagation) називають такі сучасні мережі, у яких з'єднання між вузлами утворюють певний цикл. Через такий внутрішній стан, мережа може проявляти динамічну поведінку в часі. На відміну від нейронних мереж прямого поширення, РНМ можуть використовувати свою внутрішню пам'ять для обробки довільних послідовностей вхідних сигналів. Це робить їх оптимальними для таких задач як розпізнавання несеgmentованого рукописного тексту, розпізнавання мовлення [10]. Отже, основна відмінність між ними в напрямку передачі сигналів нейронів: в МПП вихідні дані визначаються

вхідними сигналами та ваговими коефіцієнтами штучних нейронів, а в РНМ виходи нейронів визначаються не лише вагами та вхідними сигналами, а і попередніми виходами.

2.2.1 Нейронні мережі прямого поширення

Одношаровий персептрон був вперше розроблений Розенблаттом наприкінці 1950-х - на початку 1960-х років. Особливістю персептрона є здатність до навчання його вагових коефіцієнтів. Основна модель персептрона, здатного класифікувати шаблон в одному з двох класів, показана на рисунку 2.4 [4].

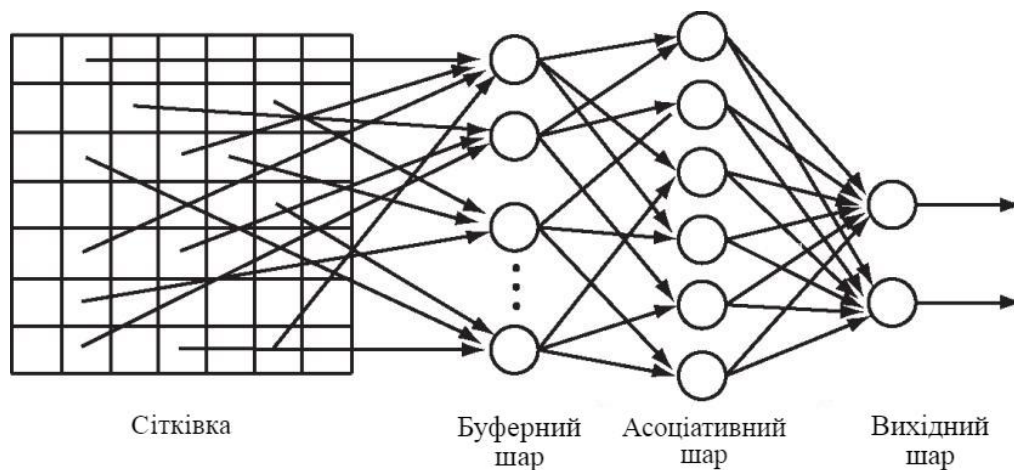


Рисунок 2.4 – Персептрон Розенблатта

Модель складається з масиву одиниць шару сітківки або сенсорів, які випадковим чином з'єднані з другим масивом асоціативних одиниць. Кожен із цих блоків продукує вихід, лише якщо активовано достатньо сенсорних одиниць, які до нього підключені, тобто вихідні сигнали асоціативних одиниць є двійковими.

Сигнали першого шару надходять на асоціативний шар, причому в загальному випадку не всі нейрони першого та другого шарів пов'язані між

собою. Зв'язки між асоціативним та вихідним шарами варіативні та можуть змінюватися в залежності від вагових коефіцієнтів. Нейрони асоціативного шару мають лінійні функції активації, тому в той час як з сітківки надходять вхідні сигнали, вони посилають нейрони на вихідний шар, де відбувається кореляція зважених імпульсів.

Другий тип моделей прямого поширення – багатошаровий перцептрон (рисунок 2.5).

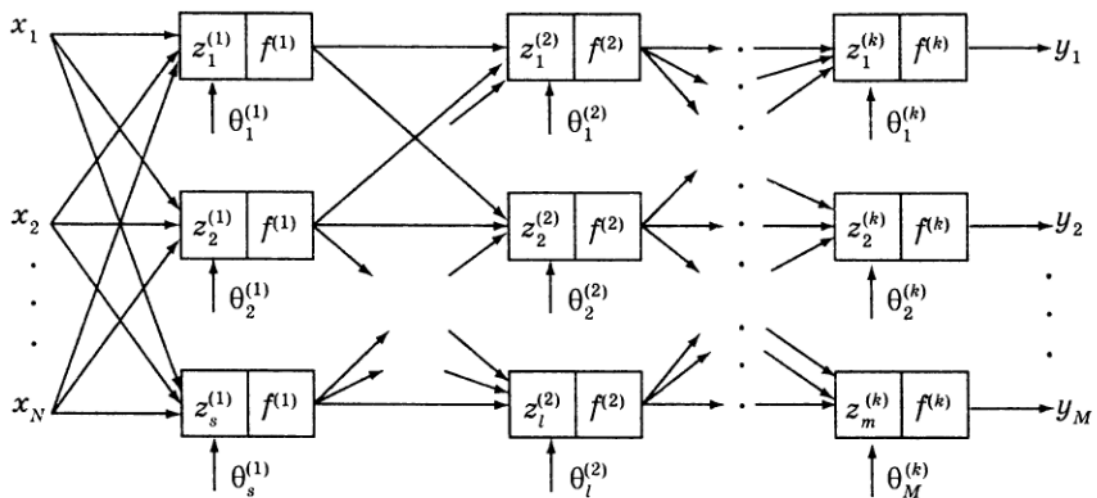


Рисунок 2.5 – Багатошаровий перцептрон

Для подолання обмежень одношарових мереж можна використовувати багатошарові мережі прямого поширення, які не тільки мають вхідні та вихідні сигнали, а також містять прихований шар, який не відноситься ні до вхідних, ні до вихідних одиниць даних. Така мережа є досить ефективною, бо використовує апроксимацію нелінійного оператора функції $f(x)$ виду:

$$\hat{y}(x) = \hat{f}(x) = f^i \left[(W^i)^T f^{i-1} \left[(W^{i-1})^T f^{i-2} [\dots f^1 [(W^1 x + b_1)^T] \dots] \right] \right] + b_i, \quad (2.2)$$

де W^i – вектор вагових параметрів нейронів i -го шару мережі;

$f^q[\cdot]$ - функція активації i -го шару;

b_i – показник зміщення i -го нейрона.

В якості функції активації застосовують функції, що представлені формулами (2.3) та (2.4):

$$f_{th}(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (2.3)$$

$$f_{log}(x) = \frac{1}{1+e^x}. \quad (2.4)$$

Варто зазначити, що питання навчання нейромережі залишається відкритим оскільки при прогнозі на основі багатошарового персептрону в якості вхідних даних виступають власні дані нейронної мережі, що призначені для майбутніх кроків прогнозу. За таких умов якість передбачення може бути низькою. Прикладами багатошарових персептронів є Мадаліна та мережа з методом групового урахування аргументів (МГУА). Мадаліна складається з багатьох мереж Адаліни, що відносяться, в свою чергу, до одношарових персептронів. За аналогією з Мадаліною, може бути побудована мережа з N Адалін, де N – певна кількість мереж [10-12].

Остання з МПП – це мережа радіальних базисних функцій (РБФ), яка використовує дані функції для активації. Вихідними даними мережі є лінійна послідовність базисних функцій входу та параметрів нейрону. Така нейронна мережа має місце коли необхідно вирішувати задачі апроксимації функцій, прогнозування за допомогою часових рядів, задачі класифікації тощо.

2.2.2 Нейронні мережі зворотного поширення

Для того щоб відобразити певну тимчасову залежність даних, часто використовуються саме рекурентні нейронні мережі (РНМ). Нейронні мережі, описані раніше багатошарові персептрони, мають фіксоване число входів і сприймають кожен з них як незалежний. У рекурентних мережах зв'язок між нейронами може надходити не тільки від нижнього шару до верхнього, але і

від нейрона до «самого себе», точніше, до попереднього значення цього нейрона або інших нейронів того ж шару. Саме це дозволяє відобразити залежність змінної від власних значень у різні моменти часу: нейрон навчається використовувати не тільки поточний вхід і те, що з ним зробили нейрони попередніх рівнів, а й те, що відбувалося з ним самим і, можливо, іншими нейронами на попередніх входах. На рисунку 2.6 проілюстровано архітектуру базової RNN.

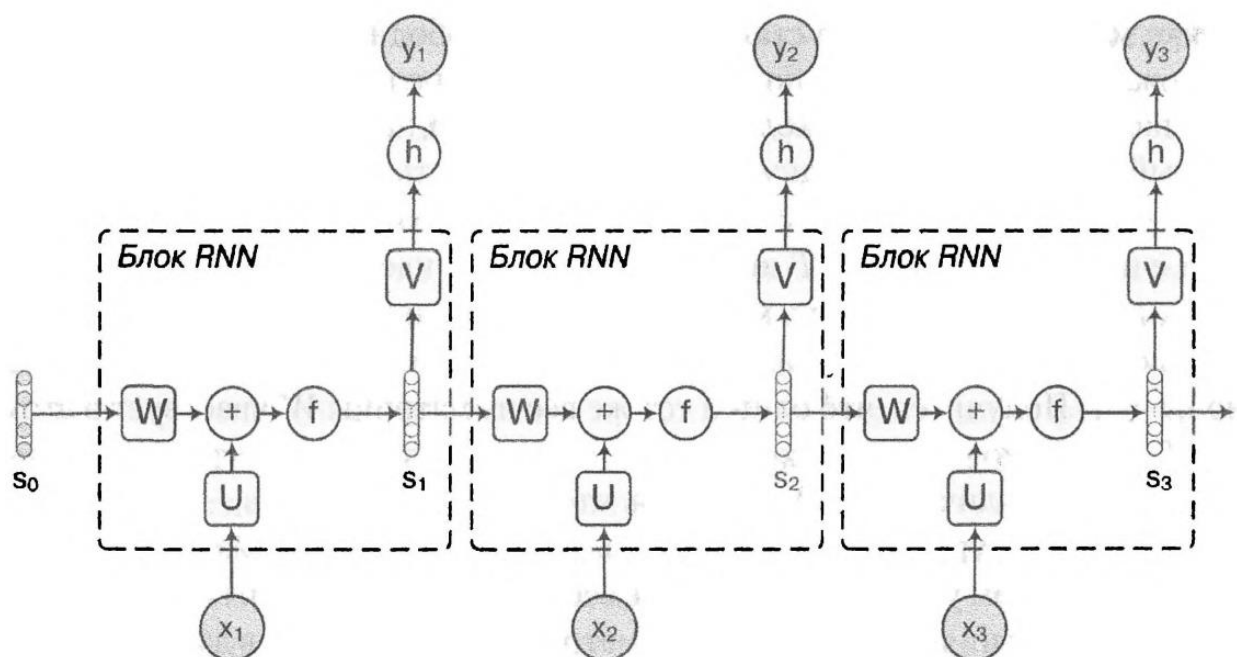


Рисунок 2.6 – Архітектура простої рекурентної нейронної мережі

Усі рекурентні мережі можна представити у вигляді такого ланцюжка повторюваних блоків нейронної мережі, які передають дані від одного модуля до іншого. Можна сказати, що на кожному кроці мережа створює кілька копій самої себе. Кожна з цих копій приймає на вхід поточне вікно в певний момент часу (певну частину послідовності) і значення, отримане з попередньої копії, потім їх комбінує і передає результат в наступний елемент. Таким чином нейронна мережа навчається. На малюнку 2.6 значення W є матрицею ваг для переходу між прихованими станами мережі. U – це матриця ваг, що подаються

на входи, а матриця V – для виходів. Входи та приховані стани вважаються векторними величинами. Будемо вважати, що на вхід блоку мережі подається завжди один вектор x . В результаті отримуємо формальну формулу (2.5) завдання мережі в момент часу x :

$$\begin{aligned} a_t &= b + Ws_{t-1} + Ux_t, s_t = f(a_t), \\ o_t &= c + Vs_t, \quad s_t = f(a_t). \end{aligned} \quad (2.5)$$

У рекурентних нейронних мережах сигнал із вихідних нейронів або нейронів прихованого шару частково передається назад на входи нейронів вхідного шару, що і називається власне зворотним зв'язком. Рекурентна мережа, така як мережа Хопфілда, фільтрує вхідні дані, повертаючись до сталого стану і таким чином дозволяє вирішувати завдання компресії даних. Мінусом рекурентної нейронної мережі є те, що вона не здатна запам'ятовувати дані надовго, але цю проблему вирішує похідна від РНМ нейронна мережа з довгою короткостроковою пам'яттю (Long short-Term Memory, LSTM) [1].

Мережею Кохонена (Когонена, самоорганізувальна мапа) називається така мережа, що має властивість самостійно організовуватися та під час надходження потоку вхідних сигналів не отримує інформацію про бажаний вихідний сигнал. Через це неможливо сформулювати критерії налаштування вагових коефіцієнтів, тому їх кореляція відбувається іншим чином. Мережа Кохонена складається з M нейронів, які утворюють шари з прямокутних решіток. У мережі Кохонена елементом вихідного шару, що виграв, є такий, у якого рівень активації найвищий, тобто він вимірює відстань від вхідного приклада до точки, координати якої зберігаються в елементі мережі.

Кожен з нейронів характеризується власним розміщенням у шарі й ваговим коефіцієнтом. Чим ближче розташовуються нейрони, тим сильніше вони впливають один на одного під час навчання. При збудженні одного нейрона можна дізнатися силу збудження інших, що знаходяться поруч з ним.

Мережа Кохонена може застосовуватися для розвитку мережі векторного квантування [12]. На рисунку 2.7 показано модель мережі Кохонена [4].

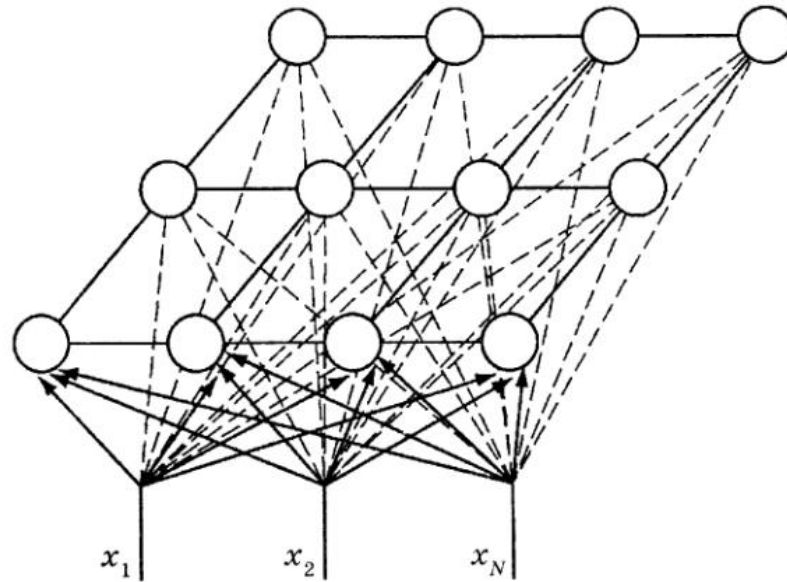


Рисунок 2.7 – Мережа Кохонена

Наступна мережа, названа на честь американського фізика Гопфілда, який поєднав ідеї усіх науковців у сфері ШНМ та висунув свою ідею. Припускають, що правило Хебба привело до появи мереж Хопфілда. Це графічні імовірнісні моделі, які реалізують асоціативну пам'ять: мережа (за формою вона зазвичай являє собою двовимірну решітку) «запам'ятовує» кілька бажаних станів-асоціацій. Запам'ятовування відбувається за допомогою навчання по Хеббу, і в результаті такого навчання стани-асоціації стають локальними мінімумами енергії мережі. Ідея мереж Хопфілда полягає в тому, щоб моделювати процеси асоціативного згадування коли існує кілька варіантів асоціацій, а обирається найближча з них.

Метод змагання. На відміну від правила Хебба, в якому багато вихідних нейронів можна збуджувати одночасно, тут вихідні нейрони змагаються. А вихідний нейрон з максимальним значенням зваженої суми є "переможцем". Виходи інших вихідних нейронів встановлюються в неактивний стан. На

тренуванні змінюються лише ваги нейронів "переможець" у сенсі збільшення близькості до цього вхідного прикладу.

Модель АРТ (Адаптивна Резонансна Теорія) – вид штучних нейронних мереж, в основі яких лежить теорія адаптивного резонансу Стівена Гроссберга і Гейла Карпентера. Включає в себе моделі навчання з учителем і без учителя, які використовуються при вирішенні задач розпізнавання образів і прогнозування. Основна ідея полягає в тому, що розпізнавання образів є, в даному випадку, результатом низхідних очікувань і висхідної сенсорної інформації. Причому низхідні очікування приймають форму запам'ятованих раніше прототипів або зразків, які потім порівнюються з властивостями об'єкта, що у нього спостерігаються. Таким чином система пропонує рішення проблеми стабільності, тобто проблеми набуття нового знання без порушення вже існуючого.

Окремо варто описати згорткові нейронні мережі та похідну від неї темпоральну згорткову нейронну мережу, оскільки її було вирішено застосувати для вирішення задачі прогнозування разом з рекурентними нейронними мережами.

2.3 Згорткова нейронна мережа

Однією з найпопулярніших глибоких нейронних мереж є згорткова нейронна мережа (CNN). Згорткова нейронна мережа або Convolutional neural network (CNN) – це добре відома архітектура глибокого навчання, натхненна природним механізмом візуального сприйняття живих істот, яка до того ж використовує варіацію багат шарового персептрона. Цю назву мережа отримала від математичної лінійної операції між матрицями, що називається згорткою. За останні кілька десятиліть вона вважалася однією із найпотужніших інструментів і стала дуже популярною у літературі, оскільки здатна обробляти величезну кількість даних. Інтерес до більш глибоких

прихованих шарів останнім часом почав перевершувати ефективність класичних методів у різних областях; особливо в розпізнаванні образів.

В першу чергу варто відповісти на питання, що ж являється згорткою і як вона стосується нейронних мереж? Формально, згорткою називають лінійну операцію яка використовується для виділення ознак, де невеликий масив чисел, який називається ядром, застосовується до вхідних даних, які є масивом чисел, який називається тензором. Поелементний добуток між кожним елементом ядра та вхідним тензором обчислюється в кожному місці тензора та підсумовується для отримання вихідного значення у відповідній позиції вихідного тензора, яке називається картою ознак (рисунок 3.2).

Якщо x^1 – це карта ознак у шарі під номером l , то результат двовимірної згортки з ядром розміру $2d + 1$ і матрицею ваг W розміру $(2d + 1) \times (2d + 1)$ на наступному шарі буде таким:

$$y_{1,j}^l = \sum_{-d \leq a, b \leq d} W_{a,b} x_{i+a, j+b}^l, \quad (2.5)$$

де $y_{1,j}^l$ – результат згортки на рівні l , $x_{i,j}^l$ – її вхід, тобто вихід всього попереднього шару.

Двома ключовими гіперпараметрами, які визначають операцію згортки, є розмір і кількість ядер. Перший параметр зазвичай дорівнює 3×3 , але іноді може бути і 5×5 або 7×7 . Другий параметр є довільним і визначає глибину вихідних карт функцій.

Ядро в згортці переміщається по площині вхідних даних, виконує множення кожного з елементів вхідного сигналу, що досліджується, і в результаті видає суму у вигляді одного вихідного значення (пікселя), який поміщається на наступний шар. Таким чином основна функція згортки – це зменшення розміру вхідних даних, тобто їх стискання, та перетворення даних у новий формат.

Падинг або нульове доповнення є технікою для вирішення проблеми покриття крайніх елементів вхідного тензора, коли рядки та стовпці нулів додаються з кожного боку вхідних даних, щоб центр ядра відповідав крайньому зовнішньому елементу та залишався в площині розмірності через операцію згортки. Сучасні архітектури CNN зазвичай використовують нульове доповнення для збереження розмірів в площині, щоб залучити більше шарів. Без нульового доповнення кожна послідовна карта функцій ставала б меншою після операції згортки.

На рисунку 2.8 показано операцію згортки розміром 3×3 .

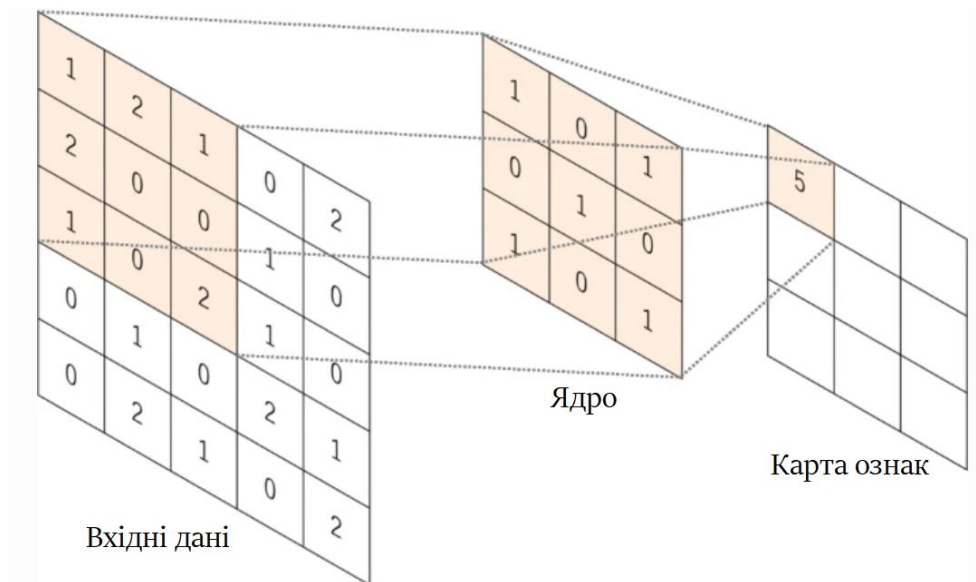


Рисунок 2.8 - Приклад операції згортки з розміром ядра 3×3

CNN має кілька шарів; а саме:

- згортковий шар;
- підвибірковий шар;
- повнозв'язний шар.

Коротко опишемо кожен з шарів та їх основні властивості й функції в згорткових нейронних мережах.

2.3.1 Згортковий шар

Окрім вхідного шару даних, згортковий шар є першим шаром, який використовується для вилучення різних функцій із вхідних зображень. У цьому шарі ми використовуємо фільтр або метод ядра, щоб витягти функції з вхідного зображення. Згортковий шар складається з набору карт ознак або матриць, в середині яких є ядро або фільтр.

На рисунку 2.9 наведено приклад згортки та формування нової матриці ваг або ядра.

3_0	3_1	2_2	1	0
0_2	0_2	1_0	3	1
3_0	1_1	2_2	2	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Рисунок 2.9 – Згортка та створення нового ядра

2.3.2 Підвибірковий шар

Підвибірковий шар також використовує карти, кількість яких співпадає з кількістю на попередньому згортковому шарі. Основна мета цього шару – зменшити розмір згорнутої карти функцій, щоб, в свою чергу, зменшити витрати на обчислення. Це виконується шляхом зменшення зв'язків між шарами та незалежної роботи над кожною картою об'єктів. Слід зазначити, що в жодному з підвибіркових шарів немає параметрів, які можна вивчати, тоді як розмір фільтра, крок і відступ є гіперпараметрами в операціях об'єднання, подібних до операцій згортки. На практиці зазвичай використовується

максимальне об'єднання з фільтром розміру 2×2 із кроком 2. Це зменшує дискретизацію площинного розміру карт об'єктів у 2 рази. На відміну від висоти та ширини, розмірність глибини карт об'єктів залишається незмінною [14].

2.3.3 Повнозв'язний шар

Повнозв'язний шар складається з ваг і зміщень разом із нейронами та використовується для з'єднання нейронів між двома різними шарами. Ці шари зазвичай розміщуються перед вихідним шаром і утворюють кілька останніх шарів архітектури CNN.

Кожен шар в мережі CNN поступово збирає уявлення про особливості об'єкта дослідження. Останні пари шарів, як правило, спеціалізуються на визначенні конкретних класів. Метод передачі навчання полягає в тому, що у CNN, яка була натренована на одному наборі даних, видаляються ваги з останнього шару для того, щоб знову тренувати цей шар на іншому наборі даних. Це дозволяє використовувати одну й ту саму згорткову нейронну мережу для класифікації різних об'єктів. Існує декілька підходів передачі навчання в залежності від схожості навчальних даних та величини тренувальної множини. Внаслідок використання цього методу, час тренування значно зменшується, тому передача навчання є корисним інструментом, коли даних недостатньо, або тренування потребує надто багато ресурсів [14].

2.4 Темпоральна згорткова мережа для вирішення задачі прогнозування

Незважаючи на те, що згорткові нейронні мережі (CNN) зазвичай асоціюються із завданнями класифікації зображень, вони виявилися цінними інструментами для моделювання та прогнозування послідовності за умови правильних модифікацій. У даній кваліфікаційній роботі досліджуються основні складові темпоральної згорткової мережі (TCN), і те, як вони

поєднуються разом для створення потужної моделі прогнозування. Бібліотека Darts, яка використовується в роботі, підтримує модуль для навчання мережі TCN, що робить прогнозування можливим без особливих навичок програмування. Модель TCN також порівнюють з канонічними рекурентними архітектурами, такими як LSTM і GRU.

Особливістю даної згорткової мережі є те, що вона задовольняє два основні принципи:

- можливість приймати послідовність будь-якої довжини та виводити її як послідовність тієї ж довжини, що й на вході (подібно до мереж RNN);
- запобігання витоку інформації з майбутнього в минуле за допомогою причинних згорток.

Для того щоб виконувався перший принцип, TCN використовує архітектуру одновимірної, повністю згорткової мережі (one-dimensional, fully convolutional network). Тобто кожен прихований шар буде доповнено нулем, щоб підтримувати однакову довжину з вхідним шаром.

Другий принцип можливий завдяки причинним згорткам, які відрізняється від стандартної згортки тим, що операція згортки, яка виконується для отримання виходу в момент часу t , не приймає майбутні значення як вхідні дані.

Це означає, що використовуючи розмір ядра k , вихідний результат o_t можна отримати з використанням значень $x_{t-(k-1)}, x_{t-(k-2)}, \dots, x_{t-1}, x_t$. Нульове доповнення довжиною $k - 1$ використовується на кожному шарі, щоб підтримувати ту саму довжину, що й вхідна послідовність [2].

Усі ці характеристики роблять TCN досить сприятливою архітектурою глибокого навчання для вирішення складних проблем часових рядів. Основною перевагою TCN мереж полягає в тому, що, подібно до RNN, вони можуть обробляти вхідні дані змінної довжини шляхом переміщення одновимірного причинно-наслідкового згорткового ядра.

На рисунку 2.10 зображено типовий приклад роботи темпоральної згорткової мережі з трьома складеними блоками даних.

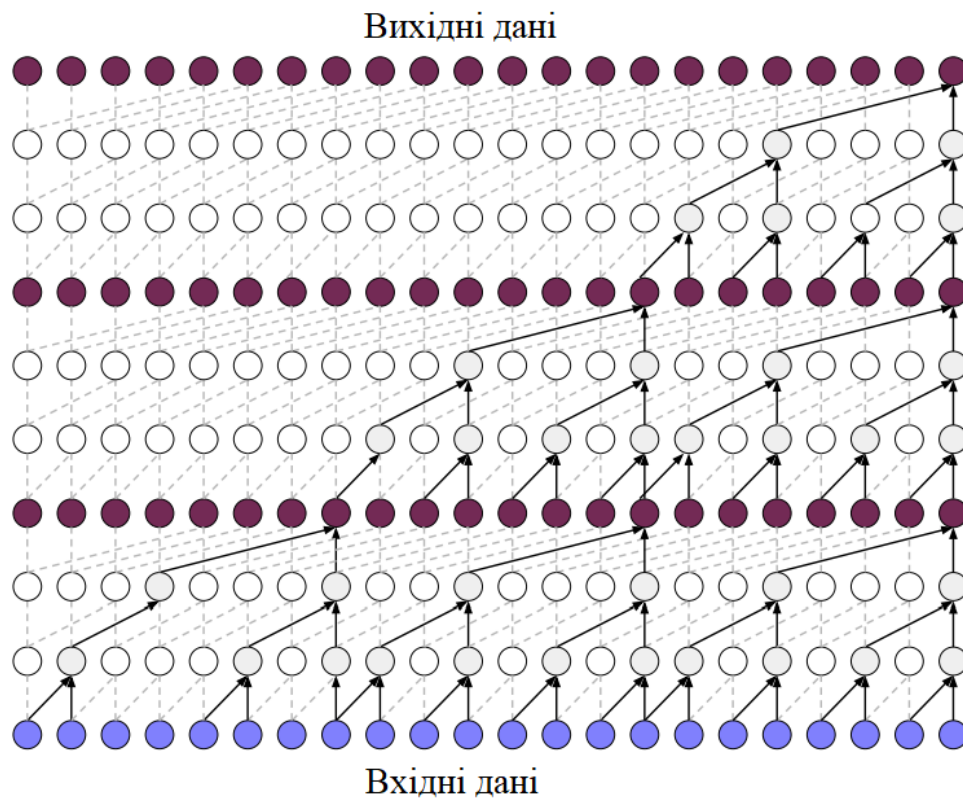


Рисунок 2.10 – Модель темпоральних згорткових мереж (TCN) із 3 складеними блоками

Крім того, TCN більш ефективно використовують пам'ять, ніж рекурентні мережі, завдяки спільній архітектурі згортки, яка дозволяє їм обробляти довгі послідовності паралельно. У RNN вхідні послідовності обробляються послідовно, що призводить до збільшення часу обчислення. Крім того, темпоральні згорткові мережі навчаються за стандартним алгоритмом зворотного поширення, таким чином уникаючи проблем градієнта алгоритму зворотного поширення через час, який використовується в RNN [2, 10].

Визначення ефективності мереж TCN для вирішення задачі прогнозування рівня CO₂ в атмосфері буде здійснюватися за рахунок порівняння отриманих результатів з базовою рекурентною мережею. Враховуючи високу складність моделей глибокого навчання, пошук

оптимальних значень для гіперпараметрів цих мереж є дуже складним завданням. У таблиці 1.1 даної роботи було наведено критерії точності прогнозування, а саме MAE, RMSE, MAPE, SMAPE. Дані показники використано щоб оцінити якість отриманого прогнозу, а також порівняти параметри з усіх обраних методів та зробити висновки про те яка ж модель є найбільш вдалою для створення прогнозу на майбутнє [14].

2.4.1 Переваги та недоліки темпоральної згорткової нейронної мережі у порівнянні з RNN

Завершити аналіз TCN варто з переліком основних переваг, а також недоліків даної мережі. Отже, нижче наведені плюси ШНМ, що розглядається:

- низький рівень потреби в пам'яті для навчання. У TCN фільтри спільно розподілені по рівнях мережі, в той час як шлях зворотного поширення залежить лише від глибини мережі. На відміну від TCN, рекурентні НМ можуть легко використовувати багато пам'яті для зберігання проміжних результатів одразу для кількох комірок у випадку довгої вхідної послідовності;

- можливість змінювати розмір сприйнятливого поля. Зробити це можна різними способами, наприклад, стогувати розширені (причинних) згорткові шари, використовувати більші коефіцієнти розширення або збільшення розміру фільтра. Таким чином, TCN забезпечують кращий контроль над об'ємом пам'яті моделі і вони легко можуть бути адаптовані до інших областей застосування;

- паралелізм. Через використання одного і того ж фільтра на кожному з шарів моделі, процеси згортки можуть виконуватися одночасно, що є відмінністю від RNN, де обов'язковою умовою є очікування завершення попередніх згорток щоб перейти до наступних етапів прогнозування;

- стабільні градієнти. Через те, що TCN мають шлях зворотного поширення мережі, це дає перевагу перед рекурентними мережами, де використовується тимчасовий напрямок послідовності. На відміну від RNN,

темпоральна згорткова мережа уникає емпіричної оцінки загальних згорткових та рекурентних мереж;

- підтримка вхідних даних змінної довжини. Модель TCN окрім того, що рекурентно моделює вхідні дані зі змінною довжиною (подібно до RNN), також може приймати вхідні дані будь-якої довжини, пересуваючи одновимірні згорткові ядра. Це означає, що TCN можуть бути використані як додаткова заміна RNN для роботи з послідовними даними довільної довжини [15-16].

Мінусами темпоральної згорткової мережі є:

- використання великого об'єму пам'яті для збереження даних протягом аналізу. В TCN потрібно зберігати необроблені дані до моменту досягнення ефективної довжини історії, що вимагає більше пам'яті.

- можлива зміна параметрів для перенесення області застосування. Таким чином, при перенесенні моделі з області застосування, де потрібен лише невеликий обсяг пам'яті, до домену, що потребує великих об'ємів пам'яті, TCN може працювати гірше через відсутність достатнього рецептивного поля [16].

2.3 Основи навчання нейронної мережі

Як було зазначено у розділі 2, здатність нейронної мережі до прогнозування часових рядів безпосередньо впливають з її здатності узагальнювати та виділяти приховані залежності між вхідними та вихідними даними. Після навчання мережа здатна передбачити майбутнє значення рівня часового ряду на основі кількох попередніх значень. Моделювання часових рядів у рамках нейромережевого підходу зводиться до завдання найкращої апроксимації нелінійної функції від багатьох змінних за набором прикладів, заданих історією часового ряду:

$$\hat{y}_{k+1} = \varphi(y_k, \dots, y_{k-n+1}) + \varepsilon_{k+1}, \quad (2.6)$$

де \hat{y}_{k+1} – прогнозоване значення рівня часового ряду;

y_k, \dots, y_{k-n+1} – значення рівнів часового ряду, що спостерігаються;

$\varphi(y_k, \dots, y_{k-n+1})$ – деяка нелінійна функція, для якої нейронна мережа служить параметричною моделлю;

ε_{k+1} – помилка прогнозу;

n – порядок моделі [2, 12].

Як було згадано раніше, нейронні мережі не програмуються, вони навчаються різними способами в залежності від типу нейронної мережі та мети дослідження.

Технічно навчання полягає у знаходженні коефіцієнтів зв'язків між нейронами при мінімізації середньоквадратичного відхилення помилки ε_{k+1} . У процесі навчання нейронна мережа здатна виявляти складні нелінійні залежності між вхідними даними та вихідними, а також виконувати узагальнення. Це означає, що, у разі успішного навчання, мережа зможе повернути правильний результат на підставі даних, які були відсутні у навчальній вибірці.

Таким чином, завдання прогнозування часових рядів за допомогою ШНМ зводиться до завдання відновлення оцінки нелінійної функції $\varphi(y_k, \dots, y_{k-n+1})$ за набором прикладів, заданих історичними даними часових рядів. Цей процес реалізується у такій послідовності:

- збір даних для навчання;
- підготовка та нормалізація даних;
- вибір топології нейронної мережі;
- експериментальний вибір характеристик нейронної мережі;
- експериментальний вибір параметрів навчання;
- навчання нейронної мережі;
- перевірка адекватності навчання;
- коригування параметрів, остаточне навчання;
- вербалізація мережі для подальшого використання [3].

3 АНАЛІЗ ІНСТРУМЕНТІВ РЕАЛІЗАЦІЇ ЗАДАЧІ ПРОГНОЗУВАННЯ

В даному розділі розглядаються програмні засоби, які були використані для реалізації поставленої задачі прогнозування рівня CO₂ в атмосфері за допомогою ШНМ та часових рядів. Мова йде про середовище програмування, мову програмування, основні бібліотеки та інші допоміжні інструменти для роботи з машинним навчанням та нейронними мережами, а також застосовані в роботі моделі прогнозування часових рядів та ШНМ.

3.1 Середовище розробки

Інтегроване середовище розробки (IDE) являється програмним рішенням для розробки програмного забезпечення, в якому програміст може запускати та виконувати код, компілювати, налаштовувати свою програму, застосовувати різноманітні функції, доступні в тому чи іншому IDE. Деякі IDE, такі як IntelliJ IDEA, Eclipse, MyEclipse, NetBeans, підтримують одразу декілька мов програмування.

Для реалізації поставленої задачі була вибрана мова програмування Python, оскільки вона підходить для вирішення задач прогнозування та роботи з нейронними мережами. Створювати Python програми можна за допомогою різних інтегрованих середовищ розробки, зокрема PyCharm, Spyder, PyDev, Visual Studio Code, Eclipse та інші. Розробник сам може обрати ту IDE, яка буде зручною та легкою у користуванні, тому це індивідуально для кожного.

Інструмент Jupyter Notebook (раніше IPython Notebooks) – це веб-додаток з відкритим вихідним кодом, який можна використовувати для створення та обміну документів чи програм, що містять код, рівняння, візуалізації, текст, а також можливість одразу побачити результат виконання коду. Назва Jupyter походить від основних мов програмування, які він підтримує: Julia, Python і R. Jupyter [17].

Головна відмінність інструменту від традиційних середовищ розробки – це можливість розбивати код на частини та виконувати їх окремо незалежно одна від одної. Ще одна перевага – відображення результату відразу під виконаним фрагментом коду. Як показує дослідження репозиторіїв платформи GitHub, Jupyter Notebook являється дуже популярним в області Data Science, аналітиці даних, глибокому та машинному навчанні, нейронних мережах, а також дозволяє отримувати попередні результати кожної частини коду, будувати графіки та здійснювати іншу візуалізацію даних. Саме тому у даній кваліфікаційній роботі було вирішено використовувати цей інструмент для програмної реалізації прогнозування [17, 18].

3.2 Мова програмування Python

Python – потужна мова програмування, яка підтримує велику множину парадигм, що оптимізовані для забезпечення високої продуктивності програмістів, та якості програмного забезпечення. Дану мову програмування в основному було розроблено для акценту на читабельності коду, а його синтаксис дозволяє програмістам описувати структурні частини програми меншою кількістю рядків коду. У Python можливе повторне використання коду та модульність через підтримку власне модулів та пакетів модулів.

Історія роботи над Python почалася наприкінці 1980-х років, коли Гвідо Ван Россум у грудні 1989 року почав розробку додатків в Centrum Wiskunde & Informatica (CWI), який розташований у Нідерландах. Вважається, що мова програмування Python була випущена в 1991 році. Виявилося, що вона використовувала набагато менше коду для вираження понять, якщо порівнювати її з Java, C++ і C. При створенні мови Python основною метою було забезпечити читабельність коду та покращити продуктивність програміста.

Пройшовши довгий шлях вдосконалень, Python став найпопулярнішою мовою програмування у світі: на його основі можна створювати безліч

автономних програм і додатків в широкому різноманітті предметних областей, з широким набором інструментів та додаткових бібліотек. Не дарма перші програмні рішення в області науки про дані, глибокого навчання та, відповідно, нейронних мереж стали можливими саме на основі мови Python. Попри те, що Python виповнилося нещодавно 30 років, він став джерелом натхнення для багатьох інших мов програмування, таких як Ruby, Cobra, Boo, CoffeeScript ECMAScript, Groovy, Swift Go, OCaml, Julia тощо [17].

3.2.1 Чому варто використовувати саме Python?

Складно відповісти чітко на питання чому ж мова програмування Python набула такої популярності серед розробників, проте можна навести ряд переваг, за допомогою яких можна зрозуміти чим керувалися мільйони користувачів перед тим як обрати найкращий на їх думку засіб для реалізації поставлених задач та ідей. Нижче наведено головні фактори, на які опираються Python-користувачі:

- якість програмного забезпечення: орієнтованість Python на читабельності, узгодженості та якості програмного забезпечення в цілому, що відрізняє його від інших рішень, які орієнтовані на написання сценаріїв. Іншими словами – код має бути читабельним, повторно використовуваним та супроводжуваним, що полегшує аналіз коду в цілому, навіть якщо він написаний не вами;
- відкритий код: розробники можуть використовувати вихідний код у своїх цілях, змінювати його та поширювати свою версію Python;
- портативність програм: це дозволяє програмам, написаним на Python, функціонувати без змін на всіх основних комп'ютерних платформах, так само як на різних операційних системах;
- інтеграція компонентів: в сценаріях мови Python можна легко взаємодіяти з іншими частинами програми або додатку, над яким працює розробник, за допомогою механізмів інтеграції. Тобто основна перевага даного

фактору – це можливість зсередини коду, написаного на Python, звертатися до бібліотек C та C++, до компонентів Java та .NET, здійснювати інтеграцію з іншими фреймворками, підключатися по мережі або через порти до окремих пристроїв та здійснювати інші взаємодії, що значно розширює його можливості;

- підтримка великої кількості бібліотек: стандартна бібліотека Python величезна, що дозволяє знайти майже всі інструменти для вирішення поставленої задачі від бібліотеки NumPy для числових обчислень, до Pandas для аналізу даних, що робить користувача незалежним від зовнішніх бібліотек;

- висока ефективність: чистий об'єктно-орієнтований дизайн Python забезпечує покращений контроль процесів, а мова оснащена відмінними можливостями обробки тексту та інтеграції, а також власною структурою модульного тестування, що робить її більш ефективною та інші переваги

- можливість роботи з Python в режимі діалогового вікна [19].

Попри достатню кількість переваг, мова програмування Python має свої недоліки. Після 30-ти років користування мовою, користувачі змогли виділити єдиний вагомий мінус, а саме низьку швидкість виконання програм, що обумовлено її інтерпретованістю. Але як показує практика даний недолік проявляється під час вирішення складних та громіздких задач, що потребують багато ресурсів. На відміну від швидкості виконання, швидкість розробки, що є переважно важливішим фактором під час роботи, не відстає від показників у мов програмування низького рівня C та C++.

Завдяки стандартній бібліотеці та інтерпретатора, Python підтримує багато різноманітних пакетів, які є у вільному доступі, серед яких і необхідні для виконання поставленої задачі в даній кваліфікаційній роботі. Мова йде про бібліотеки для глибокого навчання, нейромережевого прогнозування та аналізу результатів прогнозу на основі часових рядів та якісного відображення графічних матеріалів, такі як Darts, pandas, matplotlib та інші. Детальніше розглянемо основні модулі та пакети бібліотек, що були застосовані у цій роботі [17, 19].

3.2.2 Бібліотека Darts

Darts — це бібліотека Python для легкого маніпулювання та прогнозування часових рядів. Він містить різноманітні моделі, від класичних, таких як ARIMA, до глибоких нейронних мереж. Всі моделі можна використовувати однаково, використовуючи функції `fit()` і `predict()`, подібні до `scikit-learn`, про що буде згадано пізніше. Бібліотека також дозволяє легко тестувати моделі, поєднувати прогнози кількох моделей і враховувати зовнішні дані. Darts підтримує як однофакторні, так і багатфакторні часові ряди та моделі. Моделі на основі машинного навчання можна навчити на потенційно великих наборах даних, що містять кілька часових рядів, а деякі з моделей пропонують широкую підтримку ймовірнісного прогнозування [20]. На рисунку 3.1 зображено структурну схему можливостей бібліотеки Darts.



Рисунок 3.1 – Можливості бібліотеки Darts

Розробники бібліотеки darts прагнуть спростити аналіз часових рядів і прогнозування за допомогою Python. Darts підтримує різноманітні підходи до прогнозування, починаючи від класичних статистичних моделей, таких як ARIMA та експоненціальне згладжування, а також новітні методи, засновані

на машинному та глибокому навчанні. Крім того, `darts` містить різноманітні функції, які дозволяють нам зрозуміти статистичні властивості часових рядів, а також оцінити точність моделей прогнозування. Якщо ви хочете дізнатися більше про `darts`, ви можете прочитати цей детальний вступ до бібліотеки або звернутися до офіційної документації API. Бібліотека `darts` дозволить нам створювати різні моделі прогнозування на основі набору даних CO2 Мауна-Лоа. Крім того, ми збираємося порівняти точність цих моделей і використовувати найкращу з них для прогнозування значень концентрації CO2 в атмосфері на 2023 рік [20, 21].

3.2.3 Бібліотека Pandas

Pandas – це бібліотека Python для обробки та аналізу структурованих даних, назва якої походить від "panel data" ("панельні дані"). Панельними даними називають інформацію, отриману в результаті досліджень та структуровану у вигляді таблиць. Для роботи з такими масивами даних і було створено Pandas. Вона побудована на основі структури даних, яка називається `DataFrame` і змодельована за принципом кадрів або фреймів даних середовища статистичного програмування R. Іншими словами, `DataFrame` бібліотеки `pandas` є таблицею, схожою на електронну таблицю Microsoft Excel [22].

Основна сфера застосування – забезпечення роботи в рамках середовища Python не тільки для збору та очищення даних, але для завдань аналізу та моделювання даних. Pandas пропонує великий спектр методів роботи з таблицями, зокрема, вона дозволяє виконувати SQL-подібні запити та приєднання таблиць. На відміну від NumPy, який вимагає, щоб всі записи в масиві були одного і того ж типу, в `pandas` кожен стовпець може мати окремий тип (наприклад, цілі числа, дати, числа з плаваючою точкою та рядки). Ще однією перевагою бібліотеки `pandas` є її здатність працювати з різними форматами файлів та баз даних, наприклад, з файлами SQL, Excel та CSV.

Пакет Pandas насамперед призначений для очищення та первинної оцінки даних за загальними показниками. Статистичним пакетом Pandas у повному розумінні не являється, проте набори даних типів DataFrame і Series застосовуються як вхідні в більшості модулів аналізу даних та машинного навчання (SciPy, Scikit-Learn та інших) [22].

3.2.4 Бібліотека Matplotlib

Matplotlib — бібліотека мови програмування Python для візуалізації даних двовимірною та тривимірною графікою. Matplotlib має об'єктно-орієнтований інтерфейс, тобто користувач безпосередньо взаємодіє з кожним об'єктом. За допомогою коду можна задавати будь-який елемент діаграми, у тому числі ярлики та позначки на осях.

Бібліотека включає функції створення високоякісних візуалізацій типу лінійних діаграм, гістограм, діаграм розкиду тощо. Зображення, що генеруються в різних форматах, можуть бути використані в інтерактивній графіці, в наукових публікаціях, графічному інтерфейсі користувача, веб-додатках, де потрібна побудова діаграм. Matplotlib є гнучкою та легко конфігурованою бібліотекою, яка разом з NumPy, SciPy та IPython надає можливості, подібні до MATLAB. Працюючи в Jupyter Notebook з matplotlib, можна вивести малюнок прямо в браузері за допомогою вбудованих команд `%matplotlib notebook` і `%matplotlib inline` [20].

Варто також зауважити, що Python є дуже популярною мовою програмування не лише серед звичайних користувачів, а і серед відомих організацій та корпорацій-гігантів, таких як: Google (компоненти Google сканеру та пошукової системи), Yahoo (Карти), YouTube, Mozilla, Dropbox, Microsoft, Cisco, Spotify, Quora, Facebook, Netflix, Intel та інші.

4 РОЗРОБКА СИСТЕМИ ПРОГНОЗУВАННЯ РІВНЯ ВУГЛЕКИСЛОГО ГАЗУ У АТМОСФЕРІ

Зміна клімату, безперечно, є однією з найсерйозніших проблем, з якими стикається людство, тому експерти вважають його екзистенційною загрозою нашому виду, тобто такою, яка може призвести до непоправних наслідків. Відповідно до даних кліматичних умов з відкритих джерел, збільшення частоти екстримальних погодних явищ пов'язане зі зміною клімату, і для вирішення цієї проблеми необхідно вжити рішучих заходів у всьому світі. Інакше постраждають мільйони людей, а якість їхнього життя в наступні роки значно знизиться.

Парникові гази, такі як двоокис вуглецю (CO_2) та метан (CH_4), затримують тепло в атмосфері, тим самим зберігаючи нашу планету теплою та дружньою для біологічних видів. Незважаючи на це, людська діяльність, така як спалювання викопного палива та енергетика, призводить до викиду величезної кількості парникових газів, що призводить до надмірного підвищення середньої глобальної температури Землі. За даними Океанографічного інституту Скріппса, у минулому році концентрація вуглекислого газу в атмосфері вперше в історії людства досягла максимального рівня - 416,43 ppm (частинок на мільйон). Щороку ця цифра, на жаль, збільшується. Тому перехід до стійкої глобальної економіки є необхідним, щоб ми могли пом'якшити наслідки зміни клімату та забезпечити процвітання нашого виду.

Зважаючи на актуальність проблеми, в даній кваліфікаційній роботі було прийнято рішення здійснити прогноз рівня вуглекислого газу в атмосфері на 2023 рік на основі таких моделей прогнозування часових рядів як: базова наївна модель, модель експоненціального згладжування, модель лінійної регресії, а також штучних нейронних мереж, а саме: рекурентної нейронної мережі (RNN) та темпоральної згорткової нейронної мережі (TCN). Після

цього було проведено аналіз на предмет того яка модель або метод нейромережевого прогнозування з обраних показав найкращі результати за критеріями точності прогнозу.

Набір даних для прогнозування було взято з відкритого джерела наборів даних з обсерваторії Мауна-Лоа (MLO), що знаходиться на вулкані в Гавайях. Там розміщено дослідницький центр, який займається моніторингом атмосфери з 1950 року, а його віддалене розташування забезпечує ідеальні умови для запису кліматичних даних. У 1958 році Чарльз Девід Кілінг упорядкував програму моніторингу CO₂ і почав реєструвати наукові дані про швидке збільшення концентрації CO₂ в атмосфері. Набір даних CO₂ був завантажений в Мауна-Лоа з Інституту Океанографії Скріппса і включає щомісячні значення концентрації CO₂ в атмосфері в частині мільйонів (ppm) з 1958 р. по 2022 р.

4.1 Аналіз методів часових рядів для обраної задачі прогнозування

Після того як було отримано вибірку даних про рівень CO₂ в атмосфері, варто отримати відомості про набір даних та його статистичні властивості. В процесі програмної реалізації поставленої задачі це буде можливо за допомогою різноманітних графіків, методів аналізу часових рядів та нейронних мереж.

Як вже було зазначено, серед моделей прогнозування на основі часових рядів були вибрані наступні:

- наївна модель прогнозування;
- модель прогнозування експоненціального згладжування;
- модель прогнозування лінійної регресії.

Інформація про модель визначає тип тенденцій та сезонності, що вибирається для оцінки даних часових рядів.

Наївна модель – це модель, що може передбачити значення в точці часу t як попереднє історичне значення. При створенні наївних моделей

передбачається, що деякий останній період прогнозованого часового ряду найкраще описує майбутнє цього прогнозованого часового ряду, тому в цих моделях прогноз, як правило, є дуже простою функцією від значень прогнозованої змінної в минулому. Дана модель є досить примітивною, тому не варто чекати великої точності прогнозування. Вона не лише не враховує механізми, що визначають прогнозовані дані, а й не враховує сезонних коливань та трендів.

Експоненціальне згладжування є одним з найбільш поширених прийомів, що використовуються для згладжування часових рядів, а також для прогнозування. В основі процедури згладжування лежить розрахунок експоненціальних ковзних середніх рядів, що згладжуються. Головна перевага прогнозової моделі, заснованої на експоненціальних середніх, полягає в тому, що вона здатна послідовно адаптуватися до нового рівня процесу без значного реагування на випадкові відхилення.

Лінійна регресія призначена для отримання прогнозу безперервних числових змінних. Регресія – це умовне математичне очікування безперервної залежної (вихідної) змінної при значеннях незалежних (вхідних) змінних, що спостерігаються. Лінійна регресія базується на гіпотезі, що шукана залежність – лінійна. Побудова лінійної регресії полягає у розрахунку її коефіцієнтів шляхом найменших квадратів. Незважаючи на свою універсальність, лінійна регресійна модель не завжди придатна для якісного передбачення залежною змінною. Наприклад, якщо вихідна змінна є категоріальною чи бінарною, доводиться використовувати різні модифікації регресії.

4.2 Побудова моделей прогнозування в середовищі Jupyter Notebook

У розділі 3 було зазначено, що програмну реалізацію, маніпуляцію з моделями часових рядів, навчання нейронних мереж, отримання графічних та числових результатів буде виконуватися в додатку Jupyter Notebook.

Для початку імпортуємо набір бібліотек, які потрібні для проекту, а саме: `pandas`, `matplotlib`, а також різноманітні функції з бібліотеки `darts` та `statsmodels`:

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from darts import TimeSeries
from darts.models import *
from darts.metrics import *
from darts.dataprocessing.transformers import Scaler
import logging
```

Після імпорту бібліотек потрібно завантажити обрану вибірку даних з рівнем CO2 в кадр даних `pandas` (`DataFrame`), та обов'язково застосувати до нього методи попередньої обробки даних, такі як видалення порожніх рядків, встановлення щомісячного індексу дати та часу і видалення нульових значень:

```
df = pd.read_csv('data/forecast_data.csv', header=[0],
na_values = '-99.99')
cols = [' '.join(col).replace(' ', '') for col in
df.columns.values]
df.set_axis(cols, axis = 1, inplace = True)
df['datetime'] = pd.to_datetime(df['datetime'], origin =
'1899-12-30', unit = 'D')
df.set_index('datetime', inplace = True)
df = df[['CO2[ppm]']]
df.dropna(inplace = True)
df = df.resample('M').sum()
```

Після попередньої обробки набору даних ми використовуємо функцію `plot()` для створення простого лінійного графіку часового ряду, щоб знайти тенденцію або сезонну модель:

```
df.plot(figsize=(8,5))
plt.title('Місячна концентрація CO2 ')

plt.ylabel("CO2")
plt.xlabel("Період")
plt.show()
```

На рисунку 4.1 зображено графік з набором даних про дату та рівень вуглекислого газу в атмосфері починаючи з березня 1958 року і до кінця 2022 року.

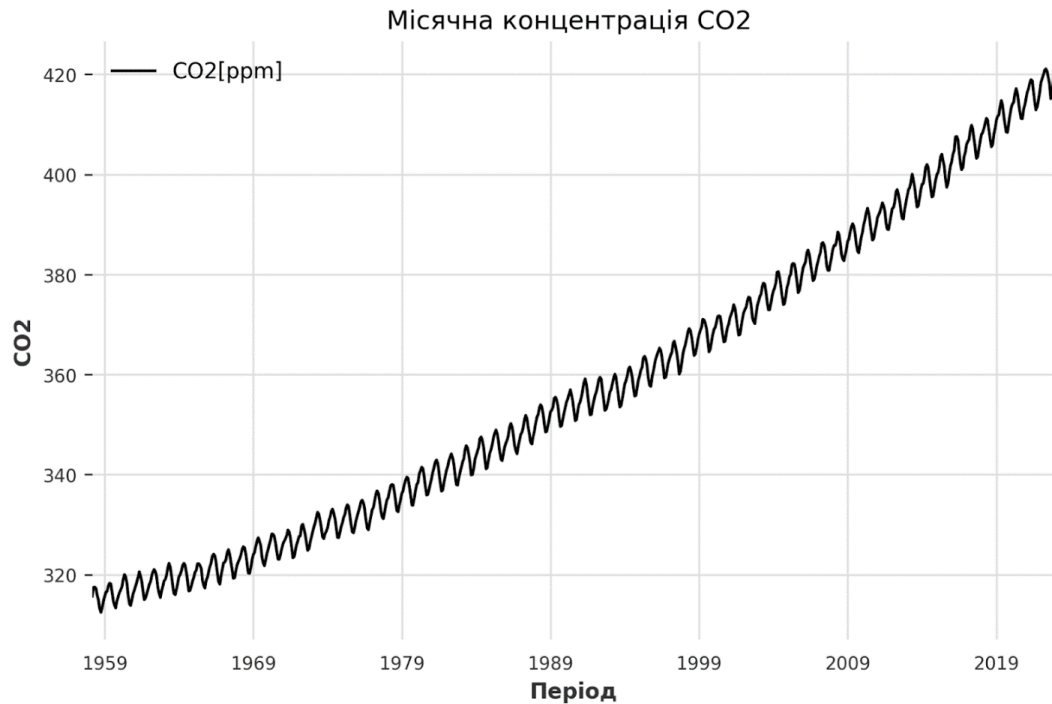


Рисунок 4.1 – Концентрація CO2 в атмосфері за період з 1958 р

З графіку на рисунку 4.1 видно, що існує чітка зростаюча тенденція, що підкреслює той факт, що концентрація CO2 в атмосфері швидко збільшувалася протягом останніх десятиліть. Крім того, існує також сезонність через природний вуглецевий цикл планети, оскільки рослини вбирають і виділяють CO2 у різні сезони року. Зокрема коли рослини починають рости навесні, вони очищають атмосферу від CO2 шляхом фотосинтезу. Відповідно навпаки, коли дерева втрачають листя восени, концентрація CO2 збільшується через дихання.

Після того як було визначено сезонність та тренд, здійснюється декомпозиція часового ряду, тобто потрібно відшукати компоненти тренду, сезонності та залишків. Для цього використовується спеціальна функція `seasonal_decompose()` бібліотеки `statsmodels`.

Після визначення компонентів, можна перенести їх на графік для кращого розуміння тенденції та сезонності, про яку згадувалось раніше:

```
mpl.rcParams['figure.figsize'] = (8, 6)
result = seasonal_decompose(df)
result.plot()
plt.show()
```

На рисунку 4.2 зображено результуючі показники тренду, сезонності, залишків завантаженого набору даних.

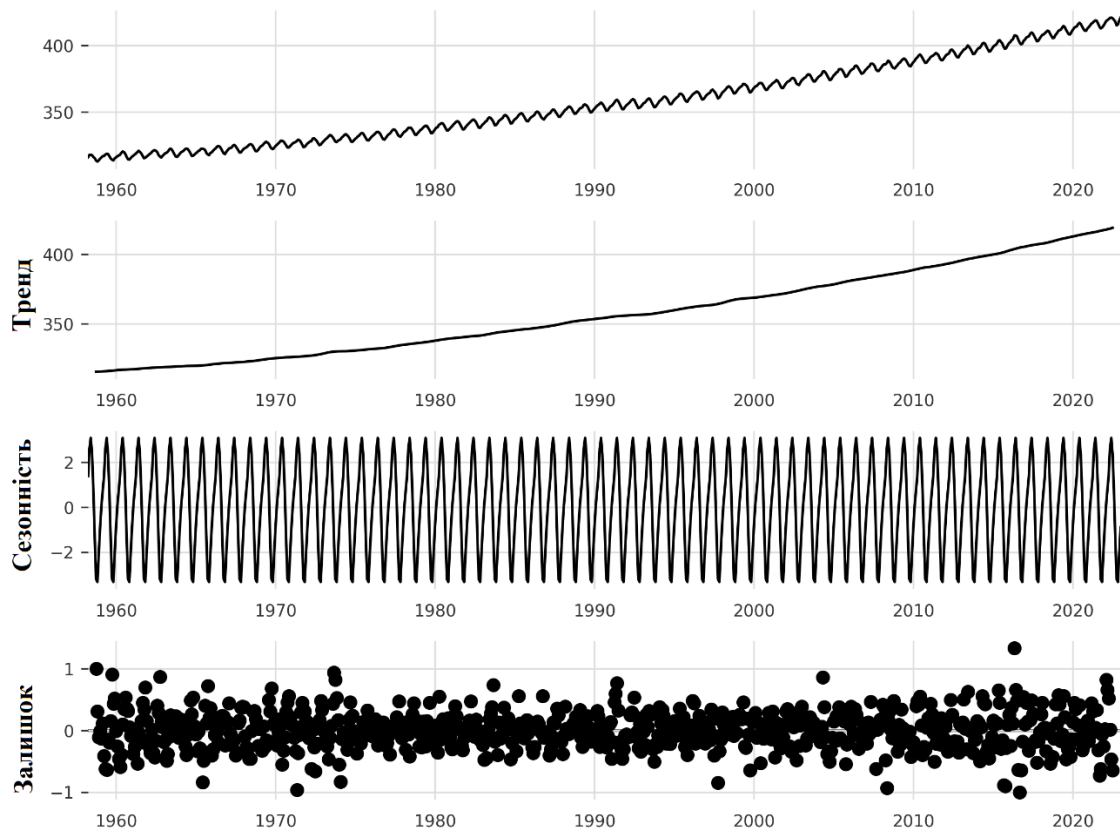


Рисунок 4.2 – Тренд, сезонність та залишки на основі даних

В даному конкретному випадку ми вже ідентифікували компоненти (сезонність, тренд) якісно аналізуючи лінійний графік (рисунок 4.1), але сезонна декомпозиція значною мірою допомагає в складніших випадках.

Наступним кроком використовується функція `plot_acf()` бібліотеки `statsmodels` для побудови автокореляційної функції часового ряду, тобто лінійної залежності між її лаговими значеннями. Автокореляція – це кореляція рівнів один з одним чи відхилень від тренду один з одним, тобто кореляція всередині того самого часового ряду, але з різними зрушеннями в часі. Автокореляція рівнів часового ряду, якщо вона істотна, свідчить про наявність тренда, тобто служить одним із методів виявлення тренду. Програмно це описується наступним чином:

```
fig, ax = plt.subplots(figsize = (8,5))
plot_acf(df, ax = ax)
plt.show()
```

Методика автокореляції складається з послідовного обчислення коефіцієнтів автокореляції відхилень із різними зрушеннями в часі. На рисунку 4.3 зображено результат роботи автокореляційної функції.

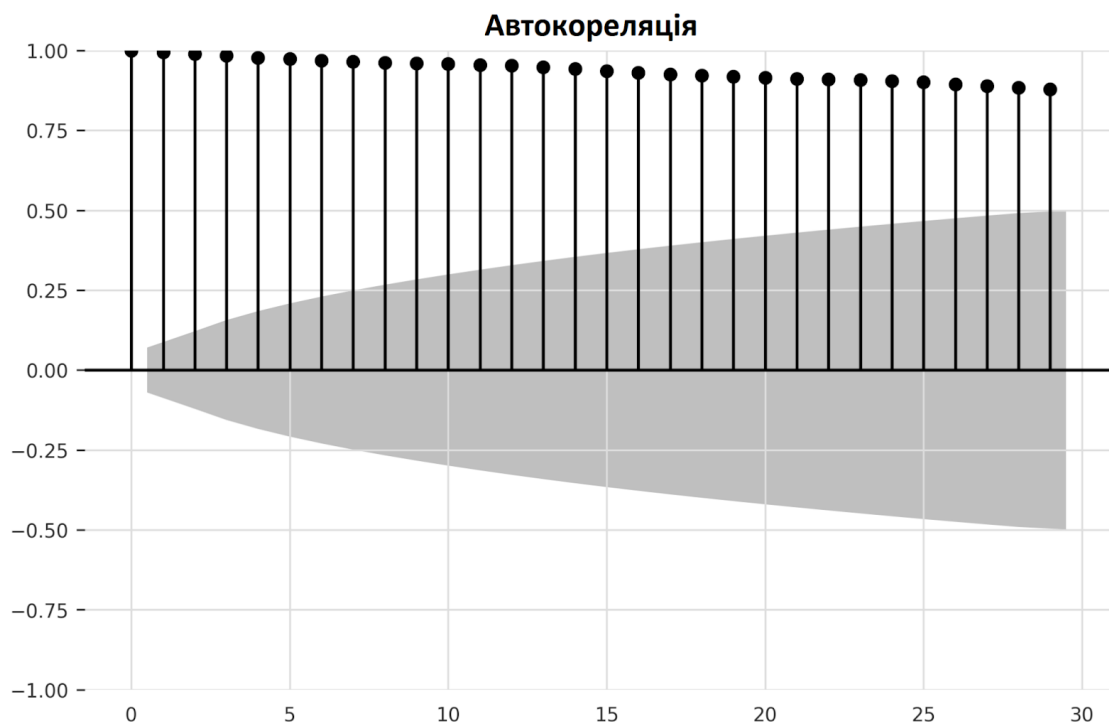


Рисунок 4.3 – Функція автокореляції

Через тренд тимчасового ряд можна бачити, що автокореляція висока для невеликих лагів і починає поступово зменшуватись після лага 5. Автокореляційна функція також повинна виділяти сезонну складову тимчасового ряду, але в цьому випадку вона непомітна.

4.3 Прогнозування часових рядів

На цьому етапі кваліфікаційної роботи буде проведено навчання обраних моделей прогнозування на основі часових рядів та нейронних мереж на завантаженому наборі даних про рівень CO₂ в атмосфері. Після отриманих результатів буде виконано порівняння їх ефективності для того щоб обрати найбільш точну модель та створити на її основі прогноз на 2023 рік.

Для оцінки точності будуть використовуватися ряд критеріїв, які згадувалися у розділі 1. Розглянемо коротко кожен з них:

- середня абсолютна помилка (Mean Absolute Error, MAE) обчислюється як середня абсолютна різниця між значеннями налаштування моделі (на один крок вперед у прогнозі прикладу) і хронологічними даними, що спостерігаються;

- середньоквадратична помилка (Mean Squared Error, RMSE) – це квадратний корінь метрики MSE. Формула відноситься до тієї ж шкали, що і значення даних, які спостерігаються. Будь-яке невелике відхилення може суттєво вплинути на показник помилки;

- середній відсоток помилок (Mean Absolute Percent Error, MAPE) є середньою абсолютною різницею у відсотках між значеннями налаштування моделі і значеннями даних, що спостерігаються та симетричний середній відсоток помилок (англ. Symmetric Mean Absolute Percent Error, SMAPE);

- коефіцієнт детермінації R^2 . Фактично, він є мірою якості моделі – це нормована середньоквадратична помилка. Якщо R^2 близький до одиниці, то модель добре прогнозує дані, якщо вона близька до нуля, то прогнози можна порівняти за якістю з лінійним прогнозуванням [14].

Почнемо процес налаштування із завантаження фрейму даних pandas в об'єкт `TimeSeries`, як того вимагає бібліотека `Darts`:

```
series = TimeSeries.from_dataframe(df)
start = pd.Timestamp('123115')
df_metrics = pd.DataFrame()
```

Додаткові функції `plot_backtest()` та `print_metrics()` дозволять побудувати прогнози та відображатимуть показники критеріїв точності моделі, а саме MAE, RMSE, MAPE, SMAPE та R^2 . Програмно цей процес описаний нижче:

```
def plot_backtest(series, forecast, model_name):
    idx = -144
    series[idx:].plot(label='Actual Values')
    forecast[idx:].plot(label='Forecast')
    plt.title(model_name)
    plt.show()
def print_metrics(series, forecast, model_name):
    mae_ = mae(series, forecast)
    rmse_ = rmse(series, forecast)
    mape_ = mape(series, forecast)
    smape_ = smape(series, forecast)
    r2_score_ = r2_score(series, forecast)
    dict_ = {'MAE': mae_, 'RMSE': rmse_,
             'MAPE': mape_, 'SMAPE': smape_,
             'R2': r2_score_}
    df = pd.DataFrame(dict_, index = [model_name])
    return(df.round(decimals = 2))
```

4.3.1 Створення наївної моделі прогнозування

Встановлення базової точності є стандартною практикою, що можна зробити на основі наївної моделі прогнозування. Це допоможе оцінити продуктивність складніших моделей, які теоретично повинні мати більшу точність порівняно з базовою. Як було описано в розділі 1, наївні методи прогнозування вважаються найпростішими або базовими з методів прогнозування. Принцип роботи моделі полягає у тому, що для оцінки якості

прогнозу використовується результуючий параметр з останнього обчислення, а всі решта попередніх значень не враховуються.

Створюємо наївну сезонну модель, яка завжди передбачає значення на k кроків назад, де k дорівнює сезонному періоду та описується програмним кодом:

```
model = NaiveSeasonal(K = 12)
model_name = 'Наївна сезонна модель'
plt.figure(figsize = (8, 5))
forecast = model.historical_forecasts(series, start=start,
forecast_horizon=12, verbose=True)
plot_backtest(series, forecast, model_name)
df_naive = print_metrics(series, forecast, model_name)
df_metrics = df_metrics.append(df_naive)
```

Для перевірки наївної сезонної моделі використовується функція `historical_forecasts()` з горизонтом прогнозу 12 місяців. Ця функція послідовно навчає модель у вікні, що розширюється, і за замовчуванням зберігає останнє значення кожного прогнозу. На рисунку 4.4 зображено результат роботи наївної моделі прогнозування на основі часових рядів.



Рисунок 4.4 – Графік прогнозу на основі наївної моделі прогнозування

Наївна модель — це тривіальне рішення, яке використовує евристику або просту статистику для обчислення прогнозу, тому розробка базової моделі далеко не завжди дає точні результати. Після створення прогнозу, можна проаналізувати отриманий графік та показники точності моделі. Як бачимо в таблиці 4.1, наївна сезонна модель спрацювала досить добре, зі значенням SMAPE 0,61%, але значення коефіцієнту детермінації є меншим 0, що вказує на те що це базова модель і результати прогнозування не задовольняють поставленої задачі.

Таблиця 4.1 – Значення критеріїв точності наївної моделі прогнозування

MAE	RMSE	MAPE	SMAPE	R^2
2.53	2.63	0.61	0.61	-0.07

4.3.2 Створення моделі прогнозування експоненціального згладжування

Тепер, коли ми встановили базову точність за рахунок наївної моделі, переходимо до налаштування наступної моделі прогнозування на основі експоненційного згладжування, класичного підходу, який успішно використовується з 1960-х років.

Як і раніше, використовуємо функцію `historical_forecasts()`, а також допоміжні функції для оцінки продуктивності моделі.

```
model = ExponentialSmoothing(seasonal_periods = 12)
model_name = 'Експоненціальне згладжування'
plt.figure(figsize = (8, 5))

forecast = model.historical_forecasts(series, start=start,
forecast_horizon=12, verbose=True)
```

```

plot_backtest(series, forecast, model_name)
df_exp = print_metrics(series, forecast, model_name)
df_metrics = df_metrics.append(df_exp)
plt.show()
df_exp

```

На рисунку 4.5 зображений результат прогнозування рівня CO₂ на основі моделі експоненціального згладжування.



Рисунок 4.5 – Графік прогнозу на основі моделі експоненціального згладжування

З таблиці 4.2 є очевидним, що модель експоненціального згладжування значно перевищує наївну модель за показником SMAPE, що дорівнює 0,11%, та за іншими якісними критеріями, як от R^2 , що прагне до 1.

Таблиця 4.2 – Точність моделі експоненціального згладжування

MAE	RMSE	MAPE	SMAPE	R^2
0.46	0.63	0.11	0.11	0.98

Також можна візуально підтвердити, що отримані прогнозовані значення є майже ідентичними до значень набору даних, нанесеним на графік (рис. 4.5).

4.3.3 Створення моделі прогнозування лінійної регресії

Моделі машинного навчання широко використовувалися при прогнозуванні часових рядів як альтернатива класичним підходам. Просто додаючи лагові значення як ознаки в набір даних, можна перетворити прогнозування часових рядів на завдання регресії. В даній роботі є можливим та доступним використання будь-якої регресійної моделі з бібліотеки `scikit-learn` або інших бібліотек із сумісним програмним інтерфейсом. В нашому випадку було створено модель лінійної регресії на основі бібліотеки `scikit-learn`.

Програмно параметри та опис для даної моделі задавалися наступним чином:

```
model = LinearRegressionModel(lags = 12)
model_name = 'Linear Regression'

plt.figure(figsize = (8, 5))

forecast = model.historical_forecasts(series, start=start,
forecast_horizon=12, verbose=True)
plot_backtest(series, forecast, model_name)
df_lr = print_metrics(series, forecast, model_name)
df_metrics = df_metrics.append(df_lr)

plt.show()
df_lr
```

Використовуючи `history_forecasts()` та інші службові функції, необхідно протестувати моделі лінійної регресії з різними параметрами, після чого можна обрати найкращу та відобразити результати. Як відомо, алгоритми

регресії намагаються знайти лінію, яка найкраще підходить для обраного набору даних. Алгоритм лінійної регресії намагається мінімізувати значення суми квадратів різниць між спостережуваним значенням і прогнозованим значенням.

На рисунку 4.6 показано отриманий результат після налаштування моделі лінійної регресії. Дана модель досить добре повторила актуальний набір даних, тому дана модель також підходить для прогнозування на майбутнє, хоча і має дещо менші показники критеріїв точності на відміну від моделі експоненціального згладжування.



Рисунок 4.6 – Графік прогнозу на основі моделі лінійної регресії

Таблиця 4.3 містить параметри точності для моделі лінійної регресії.

Таблиця 4.3 – Критерії точності моделі лінійної регресії

MAE	RMSE	MAPE	SMAPE	R^2
0.55	0.69	0.13	0.13	0.93

В цілому модель має відмінну продуктивність зі значенням SMAPE 0,13%, що трохи менше за попередню розглянуту модель, але це все одно свідчить про її потенціал.

Дана модель була останньою з обраних моделей прогнозування на основі часових рядів. Тепер розглянемо моделі прогнозування на основі нейронних мереж, а саме створимо алгоритм навчання рекурентної нейронної мережі та темпоральної згорткової нейронної мережі. Для цієї задачі застосовується бібліотека Darts, яка містить стандартні інструменти для роботи з обраними нейронними мережами.

4.3.4 Створення моделі прогнозування часової згорткової мережі (TCN)

Згорткові нейронні мережі є перспективним напрямком у машинному та глибокому навчанні, а модель TCN була створена для ще кращого аналізу даних, навчання та прогнозування. Рекурентні нейронні мережі стали стандартним вибором у дослідженнях, а також у практичних додатках. Незважаючи на це, часова згорткова нейронна модель – це альтернативна архітектура, що дає багатообіцяючі результати, тому зараз буде перевірено її продуктивність на практиці.

Навчання нейронної мережі було описано вище у даній кваліфікаційній роботі (див. 2.3), тому у першу чергу потрібно задати параметри для нейронної мережі TCN:

```
model = TCNModel(
    input_chunk_length=24,
    output_chunk_length=12,
    n_epochs=100,
    dropout=0.1,
    dilation_base=3,
    weight_norm=True,
```

```

        kernel_size=5,
        num_filters=3,
        random_state=0,
    )

```

Як і в роботі з методами прогнозування на основі часових рядів, функція `history_forecasts()` та інші службові функції використовуються для тестування моделі часової згорткової мережі та відображення результатів. Крім того, також проводиться нормалізація часових рядів за допомогою класу `Scaler()`.

Програмний код описує задані параметри та функції для навчання та виведення графічних результатів:

```

model_name = 'TCN'
plt.figure(figsize = (8, 5))
scaler = Scaler()
scaled_series = scaler.fit_transform(series)
forecast = model.historical_forecasts(scaled_series,
start=start, forecast_horizon=12,
verbose=True)
plot_backtest(series, scaler.inverse_transform(forecast),
model_name)
df_dl = print_metrics(series, scaler.inverse_transform(forecast), model_name)
df_metrics = df_metrics.append(df_dl)

```

Після ряду налаштувань та маніпуляцій з параметрами мережі, навчання та коригування результатів, отримуємо графік з прогнозом на основі моделі TCN, зображений нижче на рисунку 4.7.

За результатами критеріїв точності даної моделі (таблиця 4.4) дана модель може найкраще спрогнозувати рівень вуглекислого газу в атмосфері на майбутні роки. Вдалося досягнути ідеального показника MAPE та SMAPE, що рівні 0,1%. Решта показників теж переважають над такими, що показали моделі прогнозування на основі часових рядів.

Таблиця 4.4 – Критерії точності нейронної мережі TCN

MAE	RMSE	MAPE	SMAPE	R^2
0.41	0.63	0.1	0.1	0.96

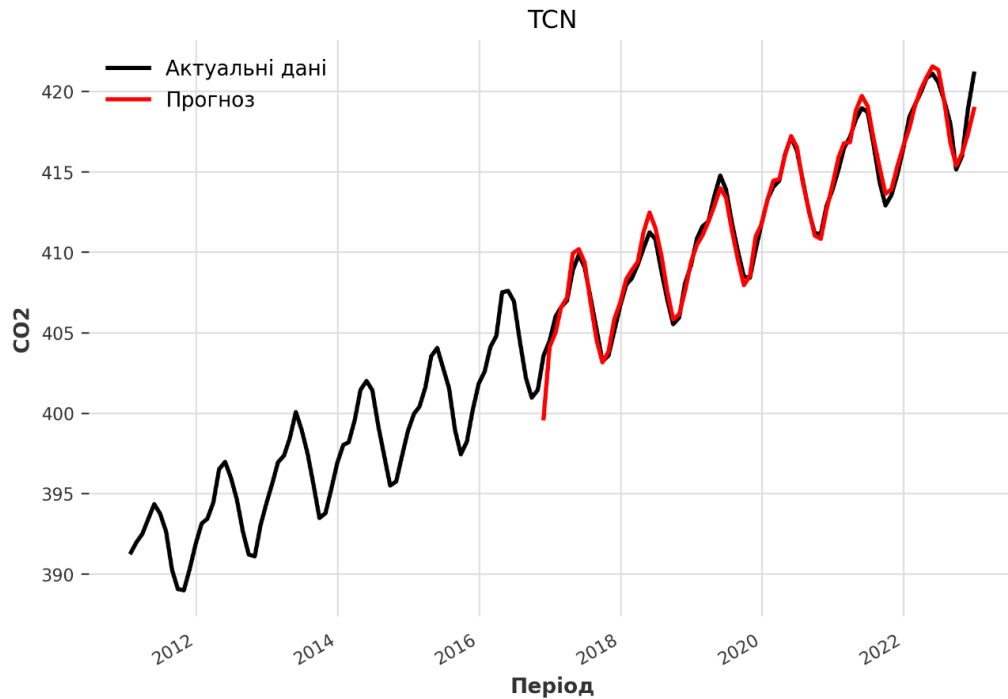


Рисунок 4.7 – Графік прогнозу на основі нейронної мережі TCN

Не можна стверджувати, що використання нейронних мереж для задач прогнозування завжди є кращим вибором, бо все залежить від поставленої задачі та мети дослідження. Кожен з методів є вдалим по-своєму. Для порівняння, проведемо також навчання рекурентної нейронної мережі.

4.3.5 Створення моделі прогнозування на основі рекурентної нейронної мережі

Мережами зворотного поширення або рекурентними називають такі сучасні мережі, у яких з'єднання між вузлами утворюють певний цикл. Через такий внутрішній стан, мережа може проявляти динамічну поведінку в часі. У рекурентних мережах зв'язок між нейронами може надходити не тільки від

нижнього шару до верхнього, але і від нейрона до «самого себе», точніше, до попереднього значення цього нейрона або інших нейронів того ж шару. Саме це дозволяє відобразити залежність змінної від власних значень у різні моменти часу.

Рекурентні нейронні мережі часто порівнюють з часовою згортковою нейронною мережею через те, що вони обидві здатні ефективно навчатися та активно використовуються в різних задачах не лише прогнозування, але і в інших областях глибокого навчання.

Процес навчання схожий з TCN. Без особливих вмінь та навичок програмування можливо отримати прогноз завдяки бібліотекам машинного навчання, в нашому випадку це darts.

Блок коду нижче описує задані параметри для навчання мережі RNN:

```
model = RNNModel(
    model="RNN",
    hidden_dim=40,
    dropout=0,
    batch_size=24,
    n_epochs=50,
    optimizer_kwargs={"lr": 1e-3},
    log_tensorboard=True,
    random_state=40,
    training_length=60,
    input_chunk_length=12,
    force_reset=True,
    save_checkpoints=True,
)
```

Процес підбирання найкращих входних параметрів рекурентної нейронної мережі відбувався за рахунок теоретичних навичок, а також за рахунок багаторазового тестування навчання мережі на тих чи інших параметрах. Іноді процес навчання мережі займав до години часу і на графіку були невдалі спроби мережі навчитися, але після оптимізації параметрів, всього за 5-10 хвилин результат був готовий. Нижче наведено програмний код з функціями для мережі RNN.


```

model_name = 'RNN'
plt.figure(figsize = (8, 5))
scaler = Scaler()
scaled_series = scaler.fit_transform(series)
forecast = model.historical_forecasts(scaled_series, start=st
art,
    forecast_horizon=12, verbos=True)
plot_backtest(series, scaler.inverse_transform(forecast), mod
el_name)
df_dl = print_metrics(series, scaler.inverse_transform(foreca
st), model_name)
df_metrics = df_metrics.append(df_dl)
plt.show()
df_dl

```

Процес навчання даної мережі відбувався не легко та потребував не малих потужностей CPU. На малюнку 4.9 зображено кінцевий результат навчання мережі RNN на вхідних даних про рівень CO₂ в атмосфері.

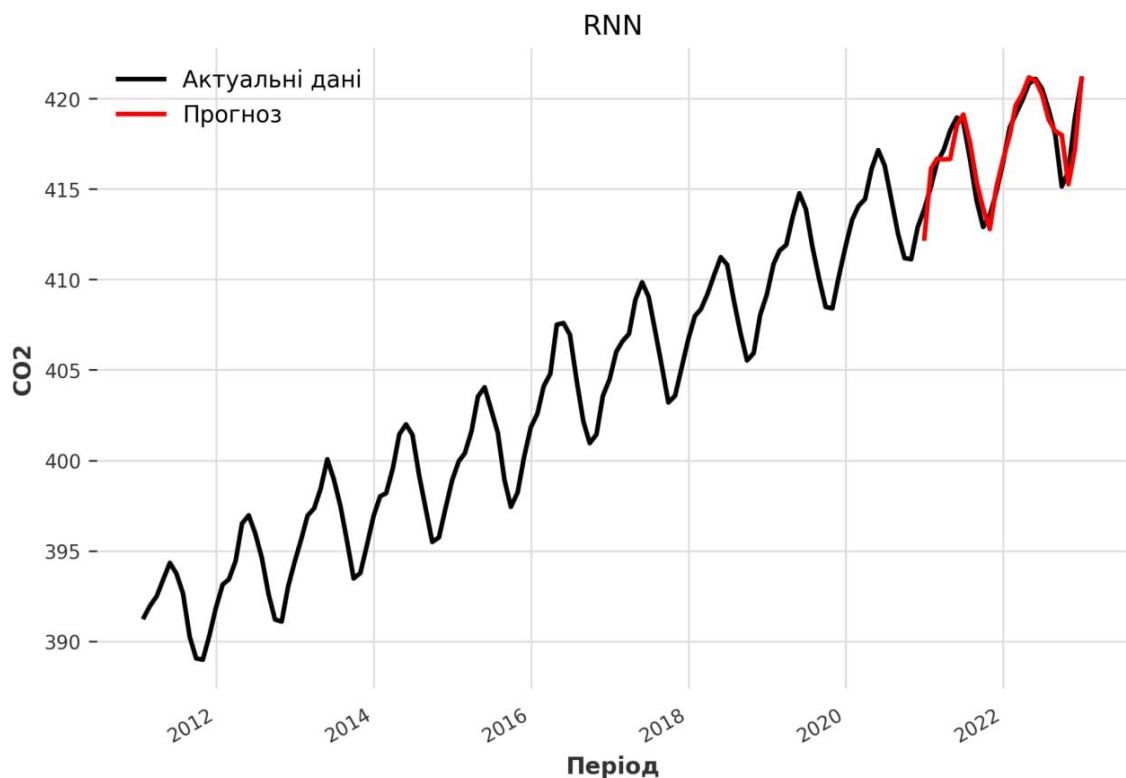


Рисунок 4.9 - Графік прогнозу на основі нейронної мережі RNN

Що стосується критеріїв точності моделі (таблиця 4.5), то вони суттєво відрізняються від TCN та експоненціального методу прогнозування.

Таблиця 4.5 – Критерії точності рекурентної нейронної мережі

MAE	RMSE	MAPE	SMAPE	R^2
0.72	0.96	0.17	0.17	0.84

З таблиці 4.5 видно, що рекурентні нейронні мережі не варто використовувати для прогнозу в даній задачі прогнозування, оскільки показник симетричної середньої відносної похибки рівний 0,17%, а коефіцієнт детермінації R^2 майже на десяту частину гірший порівнюючи з TCN та методом експоненціального згладжування.

4.4 Порівняльний аналіз отриманих результатів дослідження

Для того щоб краще зрозуміти ефективність методів, які були застосовані в даній кваліфікаційній роботі, варто зібрати результати критеріїв точності прогнозів по кожному з них у таблицю та провести порівняльний аналіз.

Зведена таблиці 4.6 містить метрики критеріїв точності прогнозів, що були використані в роботі, а саме: MAE, RMSE, MAPE, SMAPE та R^2 відповідно до кожної з моделей прогнозування: три моделі на основі часових рядів, дві моделі на основі ШНМ.

Таблиця 4.6 – Метрики критеріїв точності прогнозів по кожному із досліджуваних методів прогнозування

Метод прогнозування	Критерії точності прогнозу				
	MAE	RMSE	MAPE	SMAPE	R^2
Наївна модель	2.53	2.63	0.61	0.61	-0.07

RNN	0.72	0.96	0.17	0.17	0.84
Лінійна регресія	0.55	0.69	0.13	0.13	0.93

Продовження таблиці 4.6

Експоненціальне згладжування	0.46	0.63	0.11	0.11	0.98
TCN	0.41	0.63	0.1	0.1	0.96

Моделі та методи прогнозування відсортовані від найгіршої моделі до найкращої в рамках даного проекту. Як зазначалося раніше, наївна модель є базовою та підходить для таких задач, де є висока ймовірність повторення даних на основі недалекого майбутнього (принцип «завтра буде як сьогодні»). Вона не враховує критерії, що визначають прогнозовані дані, також не бере до уваги тренд та сезонність часових рядів, що притаманно багатьом статистичним методам передбачення.

Висока ефективність моделі TCN обумовлена її можливістю приймати послідовність будь-якої довжини та виводити її як послідовність тієї ж довжини, що й на вході, а також запобіганню витоку інформації з майбутнього в минуле за рахунок використання причинних згорток. Рекурентні нейронні мережі славляться своєю ефективністю в задачах розпізнавання несеgmentованого рукописного тексту та мовлення. Їх часто порівнюють з темпоральними згортковими нейронними мережами, проте зазвичай другі мають більше переваг над першими, що було також доведено і в даній роботі.

Моделі експоненціального згладжування також могла б бути використана для короткострокового прогнозу рівня CO₂ на майбутній рік, проте була більша практична зацікавленість провести навчання саме нейронної мережі та навчити її ефективному прогнозу.

4.5 Створення прогнозу рівня вуглекислого газу на 2023 рік

Після порівняння результатів методів прогнозування, застосованих у даній роботі (див. 4.4), було зроблено висновок використовувати мережу TCN, яка показала найкращі результати, тому короткостроковий прогноз концентрації CO₂ в атмосфері на 2023 рік буде заснований саме на ньому.

Для того щоб адаптувати модель експоненціального згладжування до всього набору даних, а потім відобразити результати, використовується функція `fit()`:

```
model_name = 'Прогноз на 2023 рік'
plt.figure(figsize = (8, 5))
scaler = Scaler()
scaled_series = scaler.fit_transform(series)
model.fit(scaled_series)
forecast = model.predict(12)
plot_backtest(series, scaler.inverse_transform(forecast),
model_name)
print(forecast.pd_dataframe())
```

На малюнку 4.10 зображено результуючий графік з прогнозом на наступний рік на основі темпоральної згорткової нейронної мережі.

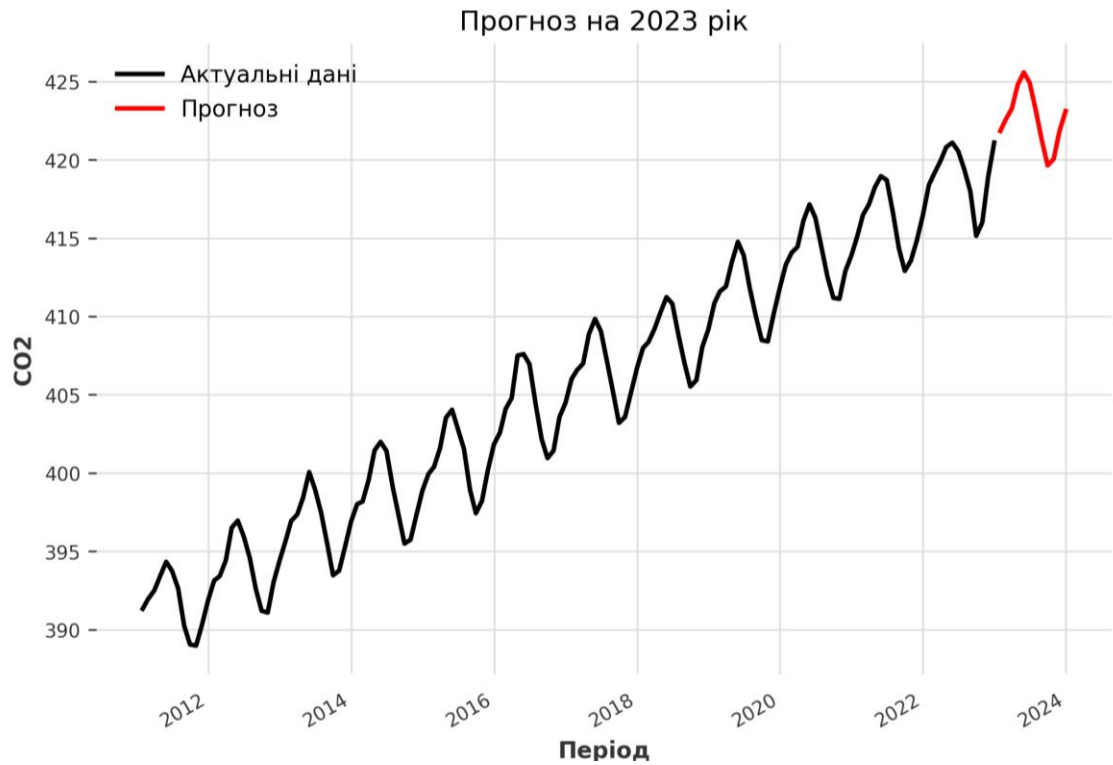


Рисунок 4.10 – Результат прогнозу рівня CO2 в атмосфері на 2023 рік

Візуально можна спостерігати, що компоненти часових рядів були успішно визначені моделлю. Таблиця 4.7 містить числові показники рівня CO2 в частках на мільйон відповідно до кожного місяця 2023 року.

Таблиця 4.7 – Прогноз на 2023 р., створений TCN моделлю

Дата у форматі рік-місяць-день	Рівень CO2, (ppm)
2023-01-31	421.836774
2023-02-28	422.576312
2023-03-31	423.262611
2023-04-30	424.778723
2023-05-31	425.581957
2023-06-30	424.922701
2023-07-31	423.185616

2023-08-31	421.267827
2023-09-30	419.635597
2023-10-31	420.052928
2023-11-30	421.851032
2023-12-31	423.111864

На жаль, рівень вуглекислого газу постійно зростає, що є причиною для хвилювання та прийняття рішучих дій для того щоб врятувати навколишнє середовище від наслідків. За допомогою налаштованої моделі прогнозування на основі часової згорткової нейронної мережі можна робити прогнози одразу на декілька років вперед. Безумовно людство має розробити стратегію для зниження показників CO₂ в атмосфері щоб запобігти подальшим змінам клімату, негативного впливу на всі живі організми та планету в цілому.

ВИСНОВКИ

Прогнозування часових рядів є областю досліджень, яка швидко набирає темпи розвитку і забезпечує багато можливостей для майбутніх робіт не лише в області прогнозування, але і в багатьох інших сферах життя. У цьому напрямку вже зроблено велику кількість цікавих та вартих уваги проектів та технологій, і в літературі запропоновано різні методи комбінування моделей прогнозування зі ШНМ.

Перевага методів ШНМ, запропонованих в цій роботі, полягає в тому, що вони забезпечують методологію для наближення до реальних даних, тобто оцінка вагового вектора не залежить від будь-якої моделі. Це звільняє людину від процедур відбору на основі моделі та припущень вибіркового даних. Коли нелінійні системи все ще перебувають у стані розвитку, є можливим зробити висновок, що підхід ШНМ пропонує конкурентоспроможний та надійний метод системного аналізу, прогнозування та контролю.

Варто відзначити, що ідеальний прогноз неможливий через наявність великої кількості факторів, які важко оцінити з високим відсотком точності. Тому, замість пошуку ідеального прогнозу, набагато важливішим є хороші знання в області існуючих моделей та правильне їх застосування в залежності від специфіки даних і предметної області, вміння пристосуватися до неідеальних прогнозів.

Через те, що прогноз залежить від минулих даних, його надійність і точність будуть знижуватися в міру того, наскільки далеко на майбутнє поставлена задача його обрахувати. Варто відзначити, що точність прогнозу і витрати на його здійснення взаємопов'язані між собою. Кращі прогнози не обов'язково є найточнішими. Такі фактори, як його мета і доступність даних грають важливу роль у визначенні бажаного рівня точності.

Динамічна поведінка більшості часових рядів у нашому реальному житті, з його авторегресивними та успадкованими термінами рухомого середнього

значення, створює труднощі для прогнозування нелінійних часових рядів, які містять успадковані середні терміни, використовуючи методології обчислювального інтелекту, такі як нейронні мережі. Даний метод моделі дозволяє визначати конкретні моделі за низьких обчислювальних витрат.

Результати показують, що темпоральні згорткові нейронні мережі переконливо перевершують не лише базові рекурентні архітектури в широкому діапазоні завдань моделювання послідовностей, але і такі методи прогнозування на основі часових рядів, як експоненціальне згладжування.

На даний момент з'являються нові підходи прогнозування часових рядів, метою яких є подолання проблем оцінки стану ринку, розмірності моделей, виявлення ознак систем вищого рівня, клімату тощо. Ці підходи базуються на застосуванні таких розділів сучасної математики, як: еволюція, теорія стохастичного моделювання, теорія катастроф і теорія систем самоорганізації, включаючи генетичні алгоритми та нечітку логіку.

Дана робота може бути корисною в перспективі для вирішення проблем надмірної кількості вуглекислого газу в атмосфері навколишнього середовища а модель прогнозу на основі TCN може бути оптимізована, за рахунок кращого налаштування гіперпараметрів. Цей метод є більш складним та потребує більше часу, але може показати значні покращення в результаті прогнозу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ratnadip Adhikari, Agrawal R. K. An Introductory Study on Time Series Modeling and Forecasting. LAP Lambert Academic Publishing, 2013. 76 p.
2. Lara-Benítez P., Carranza-García M., Luna-Romera J. M., Riquelme J. C. Temporal Convolutional Networks Applied to Energy-Related Time Series Forecasting. *Applied Sciences*. 2020. Vol. 10, no. 7, p. 2322.
3. Афанасьев В. Н., Юзбашев М. М. Анализ временных рядов и прогнозирование : учебник. М.: Финансы и статистика, 2001. 228 с.
4. Руденко О. Г., Бодянський Є. В. Штучні нейронні мережі : навч. посіб. Харків: ТОВ «Компанія СМІТ», 2006. 404 с.
5. Пасічник К. Ю. Прогнозування часових рядів із застосуванням штучних нейронних мереж. *Радіoeлектроніка та молодь у XXI столітті* : матеріали 25-го Міжнар. молодіжн. форуму, 20-22 квітня 2021 р. – Харків : ХНУРЕ, 2021. – Т. 5. – С. 189–190.
6. Song G., Li H., Witt S.F. Recent developments in econometric modeling and forecasting. *Journal of Travel Research*. 2005. Vol. 44, 82–99.
7. Archer, B. H. Demand forecasting and estimation. In J. R. B. Ritchie, & C. R. Goeldner (Eds.). New York: Wiley. 1987. 75–85.
8. L.J. Cao, Francis E.H. Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting. *IEEE Transaction on Neural Networks*. 2003. Vol. 14, 1506-1518.
9. C. Müller i S. Guido, Introduction to machine learning with Python: a guide for data scientists. 1st ed. Sebastopol: CA: O'Reilly Media, Inc, 2016, 398 p.
10. O. Rudenko, O. Bezsonov, i O. Romanyk, Neural network time series prediction based on multilayer perceptron. *Development Management*. 2019. Vol. 17, vol. 1. P. 23–34.
11. Николенко С., Кадури́н А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. Санкт-Петербург, 2018. 480 с.

12. Созыкин А. В. Обзор методов обучения глубоких нейронных сетей. Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. 60–75.
13. How to Build a Tensorflow Training Pipeline. *Medium: Towards Data Science*. URL: <https://medium.com/towards-data-science/how-to-quickly-build-a-tensorflow-training-pipeline-15e9ae4d78a0> (дата звернення: 15.11.2022).
14. Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб. Житомир : Вид. О. О. Євенок, 2020. – 184 с.
15. Bai S., Kolter J. Z., Koltun V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv, Arp. 19, 2018. (дата звернення: 01.12.2022). Доступний у <http://arxiv.org/abs/1803.01271>.
16. Smith J. E. A Temporal Neural Network Architecture for Online Learning. arXiv, Feb. 22, 2021. (дата звернення: 01.12.2022).
17. Ф. Шолле, Глубокое обучение на Python. Питер, 2018. 400 с.
18. T. Rashid, Make Your own neural network. s.l.: CreateSpace Independent Publishing Platform, 2016. 274 p.
19. Лутц М. Изучаем Python. 5-е изд. Санкт-Петербург: Диалектика, 2020. 713 с.
20. Darts: Time Series Made Easy in Python. URL: <https://unit8co.github.io/darts/> (дата звернення: 01.12.2022).
21. NOAA: Global Monitoring Laboratory. URL: <https://gml.noaa.gov/> (дата звернення: 23.11.2022).
22. Pandas: User guide. URL: https://pandas.pydata.org/docs/user_guide/ (дата звернення: 01.12.2022).